

DIPLOMARBEIT

Martin Rauchbauer

Reparaturdatenbank –

Kundenreparaturverwaltung mittels LabVIEW™

Mittweida, 2010

DIPLOMARBEIT

**Reparaturdatenbank –
Kundenreparaturverwaltung mittels LabVIEW™**

Autor: Martin Rauchbauer

Studiengang: Technische Informatik

Seminargruppe: KT05wNA

Matrikel Nummer: 17199

Erstprüfer: Prof. Dr.-Ing. Jürgen Ruck

Zweitprüfer: Dipl.-Ing. (FH) Michael Szivatz

Betreuer: Ing. Gerhard Hanifl

Eisenstadt, 16. Februar 2010

Bibliografische Angaben

Rauchbauer, Martin:

Reparaturdatenbank–Kundenreparaturverwaltung mittels LabVIEW™

74 Seiten, 55 Abbildungen, 4 Tabellen, 14 Anlagen

Hochschule Mittweida (FH), Fachbereich Informationstechnik & Elektrotechnik

Diplomarbeit, 2010

Referat:

Aufgrund steigender Kundenreparaturzahlen in einer Fertigung wurde es erforderlich, die Verwaltung der damit zusammenhängenden Daten effizienter zu gestalten. Die Idee hierzu war eine Datenbank für die Erfassung der Rückläufe in das Werk zu entwickeln.

Diese Diplomarbeit soll die Herangehensweise, technische Umsetzung und Einführung dieser Aufgabenstellung dokumentieren.

Erklärung

Ich erkläre, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

7000 Eisenstadt, 16.02.2010

Bearbeitungsort, Datum

A handwritten signature in purple ink, appearing to read 'Markus Raude', written over a dotted line.

Unterschrift

Inhaltsverzeichnis

Vorwort		3
1	Einleitung	4
1.1	Problemformulierung.....	4
1.2	Kapitelübersicht	5
1.3	Abkürzungsverzeichnis	6
2	Zielsetzung der Arbeit	7
3	Theoretische Grundlagen von Datenbanken	8
3.1	Definitionen	8
3.2	Datenbanksysteme	10
3.3	Verschieden Arten von Datenbanksystemen	12
3.4	Datenbankmanagement-Funktionen	13
3.5	Datenbankmodelle	16
3.5.1	Implementative Datenbankmodelle	16
3.5.2	Konzeptionelle Datenbankmodelle	18
4	LabVIEW™	20
4.1	Komponenten eines VIs	21
4.2	Verwendete Programmierwerkzeuge	24
4.2.1	Palette mit Bedienelementen.....	24
4.2.2	Funktionenpalette	26
4.2.3	Werkzeugpalette.....	27
5	Anforderungen an die Reparaturdatenbank	28
6	Struktur des Programms	29
6.1	VI-Hierarchie der Reparaturdatenbank	30
6.2	Kurzbeschreibung der VIs und Sub-VIs	31

7	Reparaturdatenbank	32
7.1	Ersteingabe neuer Reparaturdaten	32
7.1.1	Programmablauf Neuanlage	34
7.2	Abfrage einer Ersteingabe	37
7.2.1	Seriennummernsuche	37
7.2.2	Programmablauf Seriennummernsuche	38
7.2.3	Auftragsnummernsuche	39
7.2.4	Programmablauf Auftragsnummernsuche	40
7.3	Abschluss des Bearbeitungsvorganges	41
7.3.1	Abschließende Eingabe	41
7.3.2	Speichern der Eingabe	44
7.3.3	Programmablauf Abschluss einer Eingabe	46
7.4	Zusatzfunktionen	49
7.4.1	Neuanlage Gerätenummer und Geräte name	49
7.4.2	Löschen eines Datensatzes	50
7.4.3	Reset der Anzeige auf der Bedienoberfläche	50
8	Zusammenfassung	52
8.1	Diskussion der Ergebnisse	52
8.2	Zusammenfassung und Ausblick	53
	Literaturverzeichnis	53
	Abbildungsverzeichnis	55
	Tabellenverzeichnis	57
	Anhang	58
Anhang A	Benutzeroberfläche	59
Anhang B	Blockdiagramm Benutzeroberfläche	60
Anhang C	Blockdiagramm Auftragsnummer Suche	61
Anhang D	Blockdiagramm Endzeitberechnung	62
Anhang E	Blockdiagramm Fehlerausgabe	62
Anhang F	Blockdiagramm Fehlerausgabe Abschluss	63
Anhang G	Blockdiagramm Fehlereingabe	64
Anhang H	Blockdiagramm Fehlertext auslesen	65

Anhang I	Blockdiagramm Gerätemummernauswertung.....	65
Anhang J	Blockdiagramm Gerätemummernauswertung.....	66
Anhang K	Blockdiagramm Gerätemummernneuanlage.....	66
Anhang L	Blockdiagramm Seriennummernsuche.....	67
Anhang M	Blockdiagramm Suchen.....	68
Anhang N	Blockdiagramm Zeitberechnung.....	69

Vorwort

Die vorliegende Diplomarbeit wurde in der Zeit von September 2009 bis Februar 2010 für den Fachbereich Informationstechnik & Elektrotechnik der Hochschule Mittweida (FH) angefertigt.

Ich möchte mich ausdrücklich bei all jenen bedanken, die an der Entstehung dieser Arbeit mitgewirkt haben, insbesondere bei Herrn Prof. Dr.-Ing. Jürgen Ruck, bei meinem Zweitprüfer Dipl.-Ing. (FH) Michael Szivatz und dem betrieblichen Diplomarbeitbetreuer Herrn Ing. Gerhard Hanifl. Weiters möchte ich mich besonders bei meiner Frau Christine bedanken, die mich während meines Studiums liebevoll unterstützt hat.

1 Einleitung

1.1 *Problemformulierung*

Das Unternehmen **KEYTRONIX**, Gesellschaft für industrielle Elektronik und Informationstechnologie mbH mit Sitz in 1030 Wien, Ungargasse 64-66/1/109, sowie deren Elektronikproduktion in 7011 Siegendorf/Burgenland, Gewerbezentrum Ost, Dienstleistungszentrum 5, beschäftigt sich mit der Entwicklung, Herstellung sowie Vertrieb von Individuallösungen im Bereich Telekommunikation und Internet-Service-Providern.

Im Zuge von Rücksendungen von Geräten aus der Serienproduktion (Garantiefälle, kostenpflichtige Reparaturen) ergab sich die Anforderung seitens der Produktionsleitung, die Verwaltung der einzelnen Kundenreparaturaufträge zu vereinfachen und zu automatisieren, um eine Nachverfolgbarkeit relevanter Daten wie z. B. fehlerhafte Bauteile, Serienfehler, Durchlaufzeit und wiederholten Fehlerfall einfacher erkennen bzw. auswerten zu können. Diese genannten Daten wurden bis dato in Form von Excel-Dokumenten aufgezeichnet und abgelegt. Von der Geschäftsführung wurde an das Qualitätsmanagement die Aufgabe herangetragen eine Datenbank erstellen zu lassen, die diese Features beinhalten soll. Wenn möglich sollten keine externen Firmen, sondern eigene Ressourcen für diese Aufgabe herangezogen werden. Da in der elektronischen Fertigung von Firmengründung an mit dem grafischen Programmierwerkzeug LabVIEW™ von National Instruments gearbeitet wurde, um Funktions- sowie End-Test der einzelnen produzierten Module zu automatisieren, wurde von der Testabteilung entschieden, diese Software auch für die Realisierung des Projektes zu verwenden. Der Verfasser dieser Arbeit stellte sich zur Verfügung, eine Reparaturdatenbank mithilfe des vorhandenen Programmierwerkzeugs zu entwickeln.

1.2 *Kapitelübersicht*

Kapitel 2 erörtert die Zieldefinitionen dieser Arbeit.

Kapitel 3 befasst sich mit den theoretischen Grundlagen zum Thema Datenbanken.

Kapitel 4 beinhaltet eine Einführung in das Programmierwerkzeug LabVIEW™.

Kapitel 5 beschreibt die Anforderungen an die Datenbank.

Kapitel 6 gibt einen Überblick über die Struktur der programmierten Datenbank.

Kapitel 7 zeigt die Funktionen der programmierten Datenbank sowie deren Umsetzung aus programmtechnischer Sicht.

Kapitel 8 fasst die Ergebnisse der Arbeit zusammen und gibt einen Ausblick auf weitere Verwendung sowie zusätzliche Applikationen.

In **Anhang A** wird die Benutzeroberfläche abgebildet.

Anhang B-N beinhaltet die Blockdiagramme der einzelnen Programmteile.

1.3 **Abkürzungsverzeichnis**

Abb.	Abbildung
ANSI	American National Standards Institute
API	Application Programming Interface
CD-ROM	Compact Disc Read-Only Memory
DBMS	Datenbankmanagement-System
DVD	Digital Versatile Disc
GPIB	General Purpose Interface Bus
IC	Integrated Circuit
LabView	L aboratory V irtual I strumentation E ngineering W orkbench
PCI	Peripheral Component Interconnect
RS-232	Recommended Standard 232
RS-485	Recommended Standard 485
SMD	Surface - Mounted Device
SQL	Structured Query Language
Sub-VI	Subroutine VI – Unterprogramm
TCP/IP	Transmission Control Protocol / Internet Protocol
VI	virtuelles Instrument, LabVIEW – spezifische Bezeichnung für “Programm“.
VME	Versa Module Eurocard
VXI	VME eXtensions for Instrumentation

2 Zielsetzung der Arbeit

Schwerpunkt und auch praktischer Teil dieser Arbeit ist die Realisierung und Konzeption einer Reparaturdatenbank mittels LabVIEW™. Anhand der Vorgaben der Firma Keytronix sollen die einzelnen Daten eines Reparaturvorganges gespeichert und in einer Datenbank abgelegt werden.

Es wird ein Einblick in die Grundlagen von Datenbanken gegeben und deren Aufbau beschrieben, sowie eine kurze Einführung in LabVIEW™ von National Instruments gemacht.

3 Theoretische Grundlagen von Datenbanken

3.1 Definitionen

„Eine Datenbank ist ein verteiltes, integriertes Computersystem, das Nutzdaten und Metadaten enthält“ (Geisler, 2009 S. 23).

Aus dieser Definition geht hervor, dass Daten, die der Benutzer in einer Datenbank anlegt und als Informationsquelle heranzieht, als Nutzdaten bezeichnet werden. Metadaten hingegen, auch als Daten über Daten beschrieben, dienen dazu, die Nutzdaten in der Datenbank zu strukturieren. Eine zweite Definition beschreibt eine Datenbank als geordnete, selbstbeschreibende Sammlung von miteinander in Beziehung stehenden Daten. Das Arbeiten mit Daten beschreibt man als Datenmanagement. Zu dessen Aufgaben zählen Erzeugung, Speicherung und Wiedergabe von Daten (Geisler, 2009 S. 23).

Im Normalfall werden sogenannte Datenbankmanagement-Systeme (in weiterer Folge DBMS genannt) eingesetzt, um eine Datenbank auf einem Rechner zu verwalten (siehe Abb.: 3-1). Es kümmert sich um die Organisation der Daten und regelt den Zugriff auf diese. Man unterscheidet zwischen DBMS, die aus einem einzelnen Programm bestehen wie z. B. Microsoft Access, oder aus vielen Programmen, welche durch Zusammenarbeit die Funktionalität eines DBMS bereitstellen. Letzteres wird meist bei servergestützten DBMS verwendet. Durch eine integrierte Abfragesprache kann man sogenannte Ad-hoc-Abfragen starten, um eine bestimmte, aktuell benötigte, Anfrage zu beantworten. Zum Beispiel wird ein Kundenname eingegeben und alle, mit dem Kunden verknüpften Daten werden zurückgeliefert. Hinzufügen, Ändern und Löschen von Daten wird durch ein DBMS wesentlich erleichtert. Es können leistungsfähige Suchfunktionen implementiert werden, die das Finden bestimmter Daten beschleunigen. Abbildung 3-1 zeigt den Ablauf zwischen dem DBMS und der Datenbank an sich.

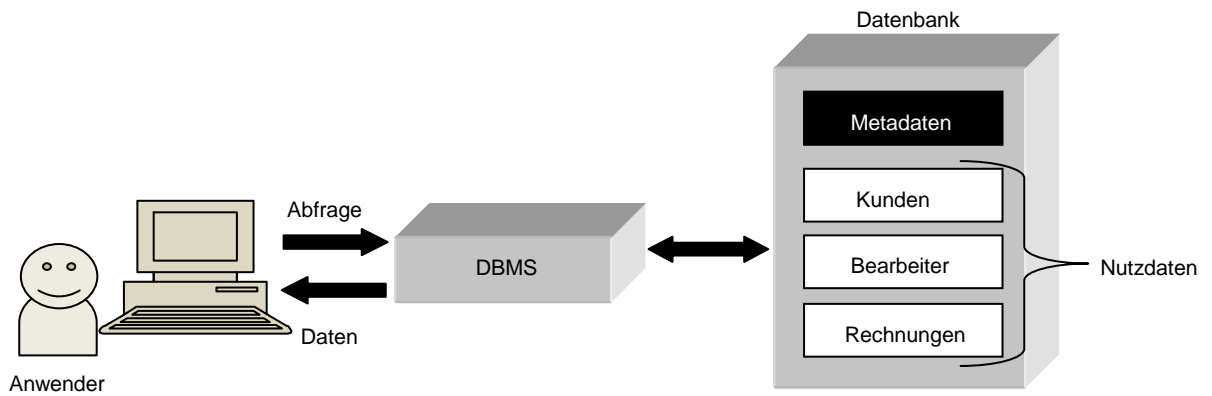


Abb. 3-1: Das DBMS verwaltet den Zugriff auf die Datenbank (vgl. Geisler, 2009, S. 24).

Der Benutzer stellt eine Anfrage an die Datenbank. Diese wird an das DBMS weitergeleitet, welche wiederum die Daten aus der eigentlichen Datenbank abfragt und an den Benutzer zurückgibt (Geisler, 2009 S. 24).

Daten, Datensätze, Datei

Daten sind alle Fakten, die in einer Datei gespeichert sind. Sie selbst besitzen einen geringen Informationsgehalt. Beispiele für Daten sind Telefonnummern, Postleitzahlen, Adressen und so weiter.

Ein Datensatz besteht aus einer Sammlung von verknüpften Feldern, die Daten über eine Person oder einen Gegenstand beherbergen. Als Beispiel kann hier ein Datensatz Daten von einem Kunden wie Adresse, Telefonnummer usw. beinhalten.

Eine Datei benennt eine Menge von Datensätzen, die zusammengehören (Geisler, 2009 S. 32).

	KUNDE	KUNDE_TELEFON	KUNDE_ANDRESSE	KUNDE_PLZ	BEARBEITER	KONTAKT_DATUM	
	Ernst Müller	0563 - 435686	Waldweg 23	97123	Hans Moser	23.09.2003	} Datei
	Hans Fuchs	0583 - 494813	Kirchengasse 45	02187	Inge Lück	23.09.2003	
	Mark Kunz	0836 - 967367	Am Graben 12	70349	Franz Zartl	23.09.2003	
	Sandra Falk	0749 - 632391	Lange Gasse 5	92375	Sabine Meyer	23.09.2003	
	Ingo Wolf	0485 - 183920	Hauptplatz 1	39815	Xaver Bauer	23.09.2003	
	Emil Schmid	0193 - 812671	Langriedgasse 98	24910	Fritz Franz	23.09.2003	

Die Tabelle ist als 'Datei' bezeichnet. Die Spaltenüberschriften sind als 'Feld' und die Zeilenüberschriften als 'Daten' gekennzeichnet. Ein Pfeil weist auf eine Zeile als 'Datensatz' hin.

Abb. 3-2: Inhalt einer Datei (vgl. Geisler, 2009, S. 32)

3.2 *Datenbanksysteme*

Mit modernen Datenbanksystemen ist es möglich, Beziehungen zwischen einzelnen Datensätzen zu überwachen. So kann ein unerwünschtes Löschen von Daten vermieden werden.

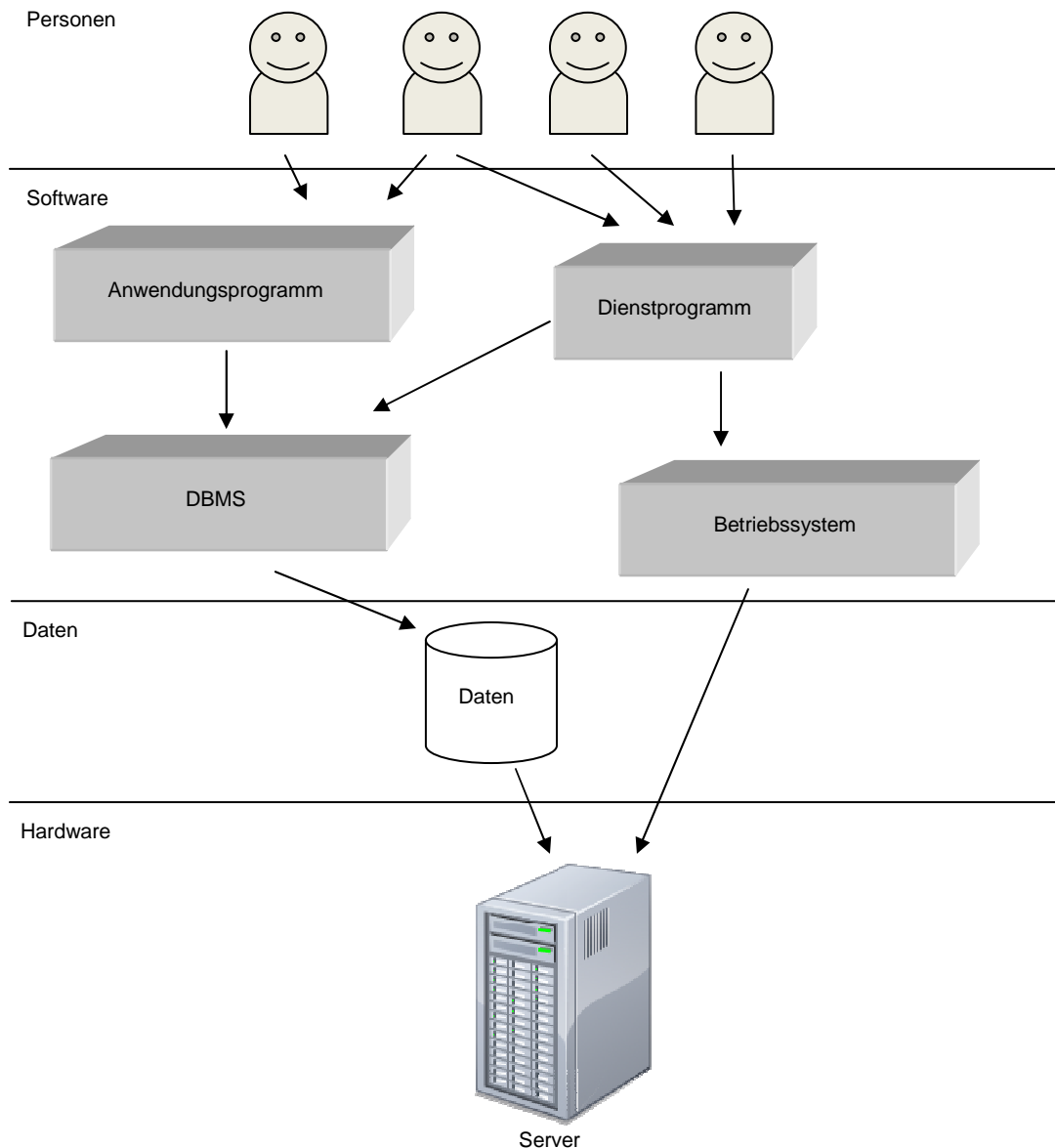


Abb. 3-3: Das Datenbanksystem (vgl. Geisler, 2009, S. 46)

Ganze Datenbanksysteme bestehen nicht nur aus dem DBMS. Es gibt zahlreiche Komponenten, welche nachfolgend kurz beschrieben werden.

Hardware

Die unterste Schicht in der gezeigten Abbildung (Abb. 3-3) stellt die Hardware dar. Sie bildet die Grundlage des Gesamtsystems. Zu dieser Komponente zählen alle elektronischen Teile wie Datenbankserver, Clientrechner, Hubs, Router, Bridges, Firewalls sowie Kabel, welche die einzelnen Geräte verbinden. Des weiteren werden auch Peripheriegeräte wie z. B. Drucker, Magnetbänder oder CD-ROMs und DVDs zur Erstellung von Sicherheitskopien zur Hardware hinzugezählt (Geisler, 2009, S. 45- 48).

Daten

Fakten, welche vom Datenbanksystem gespeichert werden, bezeichnet man als Daten. Bis auf die Speicherplatzzuordnung besitzt nur das DBMS Zugriffsrechte auf die Daten. Bei einzelnen Systemen wie etwa Oracle wird auch der Speicherplatz über das DBMS verwaltet (Geisler, 2009, S. 45- 48).

Software

Neben dem wichtigsten Teil, dem DBMS, das die Verwaltung des Datenbanksystems übernimmt, gibt es noch ein oder mehrere Anwendungsprogramme, über die der User mit der Datenbank kommuniziert. Zusätzlich gibt es noch Dienstprogramme, um Wartungsarbeiten wie z. B. Datensicherung oder Überprüfungen des Systems durchzuführen. Als letzter Teil muss hier noch das Betriebssystem erwähnt werden. Dazu gehört nicht nur jenes auf dem Server, sondern auch das verwendete Betriebssystem auf den Clientrechnern (Geisler, 2009, S. 45- 48).

Personen

Es gibt hier die Gruppe der Anwender, die mithilfe der Anwendungsprogramme Daten erfassen oder auswerten. Dann gibt es noch die Datenbankadministratoren, die das System verwalten, sowie eine fehlerfreie Funktion der Datenbank sicherstellen. Zu ihren Aufgaben gehört das Freigeben von Speicherplatz, Erteilen von Berechtigungen und Datensicherung. Zuletzt

sind da noch die Datenbankdesigner und Programmierer. Sie legen Strukturen fest und tragen die Verantwortung über die komfortable Nutzung des Systems. Die hier genannten Gruppen findet man in der Datenbank als Berechtigungsgruppen wieder. Je nach Zuteilung in eine der beschriebenen Gruppen können verschiedene Aktionen im Datenbanksystem ausgeführt werden (Geisler, 2009, S. 45- 48).

3.3 *Verschieden Arten von Datenbanksystemen*

Einzelbenutzer- oder Mehrbenutzer-Datenbank

Es kann entweder nur eine Person oder mehrere Benutzer gleichzeitig auf die Datenbank zugreifen (Geisler, 2009, S. 49-50).

Desktop-Datenbank oder Server-Datenbank

Das DBMS wird auf einem Desktoprechner (Einzelrechner) oder auf einem Server ausgeführt (Geisler, 2009, S. 49-50).

Desktop-Datenbank, Workgroup-Datenbank oder Enterprise-Datenbank

Wenige Benutzer greifen auf eine Desktopdatenbank zu, die lokal auf einem Rechner installiert ist. Bei bis zu 50 Personen spricht man von einer Workgroup-Datenbank, bei mehr Anwendern von einer Enterprise-Datenbank (Geisler, 2009, S. 49-50).

Zentrales Datenbanksystem oder verteiltes Datenbanksystem

Die Daten werden entweder auf einem zentralen Server gespeichert oder mithilfe eines Distributed DBMS auf mehrere Server verteilt (Geisler, 2009, S. 49-50).

Transaktionales Datenbanksystem oder Decision Support System (DSS)

Die Datenbank reagiert direkt auf Benutzereingaben und ein interaktives Arbeiten ist möglich, oder es erfolgen nur Ausgaben, die helfen Entscheidungen zu treffen (Geisler, 2009, S. 49-50).

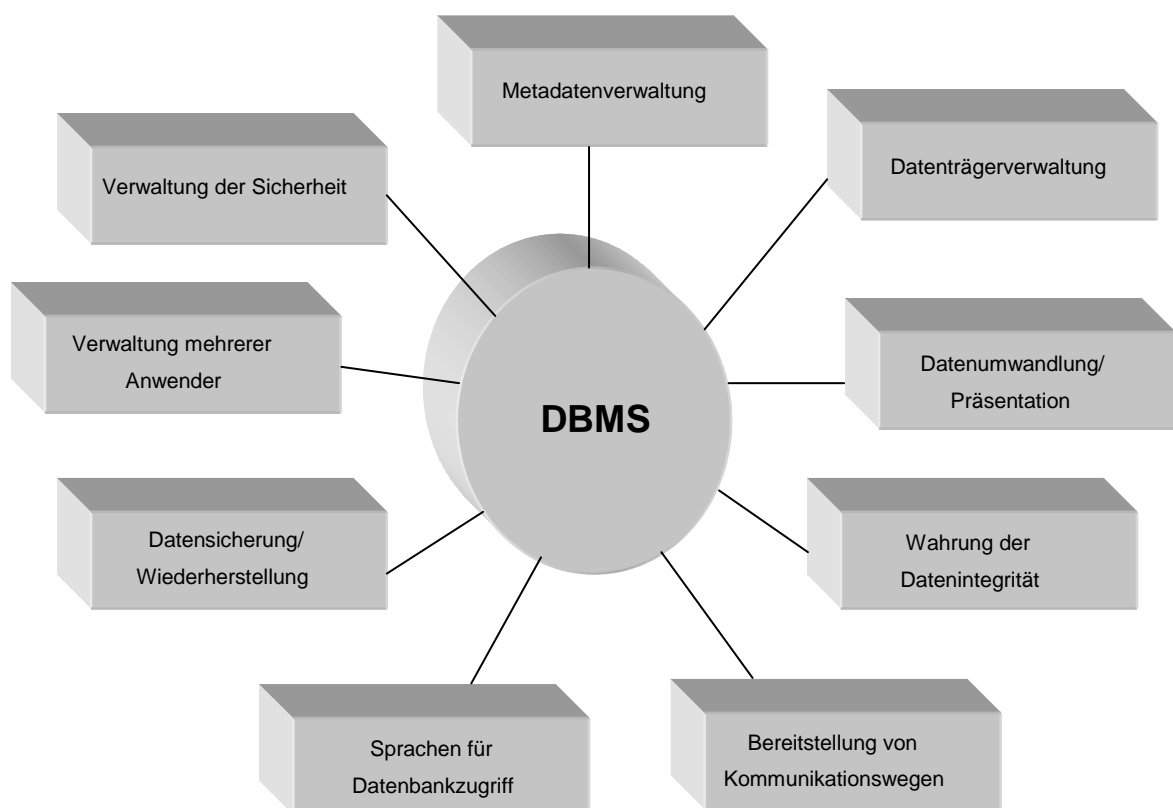
3.4 Datenbankmanagement-Funktionen

Abb. 3-4: Funktionen eines DBMS (vgl. Geisler, 2009, S 51)

Verwaltung der Sicherheit

Zugriffe und durchzuführende Aktionen auf die Datenbank werden zugelassen oder verweigert. Die Berechtigungen der einzelnen Benutzer werden verwaltet (Geisler, 2009, S. 50-53).

Verwaltung mehrerer Anwender

Da bei großen Datenbanken hunderte oder gar tausende Personen gleichzeitig zugreifen, muss für eine Datei, auf die schreibend zugegriffen wird, der Zugang für einen zweiten Benutzer gesperrt werden. Das Lesen der Datei ist jedoch uneingeschränkt von mehreren Personen gleichzeitig möglich (Geisler, 2009, S. 50-53).

Datensicherung/Wiederherstellung

Hardwarebedingte sowie durch Benutzer verursachte Fehler werden durch spezielle Tools, mit denen Datenbanksicherung als auch Wiederherstellung möglich sind, geschützt (Geisler, 2009, S. 50-53).

Sprachen für Datenbankzugriff

Die als ANSI-Standard definierte Abfragesprache ist SQL (Structured Query Language). Eingaben, die ein User über grafische Benutzeroberflächen tätigt, werden intern in SQL-Befehle umgewandelt. Zusätzlich gibt es noch sogenannte APIs (Englisch: „*application programming interface*“, Deutsch: „Schnittstelle zur Anwendungsprogrammierung“), die mithilfe von Programmiersprachen wie C++, C#, Delphi oder Visual Basic einen direkten Zugriff auf Funktionen des DBMS erlauben (Geisler, 2009, S. 50-53).

Bereitstellung von Kommunikationswegen

Um die Kommunikation des Clientrechners mit dem Datenbankserver bereitzustellen, wird im Normalfall auf herkömmliche Netzwerkprotokolle wie TCP/IP aufgesetzt. Weiters gibt es hier die so genannte Middleware, eine Softwarekomponente, die als Bindeglied zwischen Datenbank und Datenbankanwendung besteht. Durch sie ist es möglich, mit einer Anwendung auf die Datenbanksysteme verschiedener Anbieter Zugriff zu nehmen (Geisler, 2009, S. 50-53).

Wahrung der Datenintegrität

Datenintegrität setzt voraus, dass Daten konsistent und korrekt sind. Zur Wahrung dieser bieten moderne DBMS Möglichkeiten, Regeln zu definieren. Die Datenredundanz – Daten, die mehrfach im System gespeichert sind, können eventuell ohne Verlust von Informationen dezimiert werden - wird verringert und die Datenkonsistenz wird maximiert (Geisler, 2009, S. 50-53).

Datenumwandlung/Präsentation

Die Datenumwandlung benötigt man zur Trennung von logischen und physikalischen Datentypen, also Eingabeformat und Speicherformat. Als Beispiel soll hier die Eingabe bzw. Speicherung eines Datums dargestellt werden. Das Eingabeformat des Benutzers, zum Beispiel 01.01.2000, wird im System als physikalische Zahl nach dem julianischen Datum (nicht mit dem julianischen Kalender zu verwechseln!) mit dem Wert 2.451.545,0 gespeichert. Durch diese Vorgehensweise erreicht das DBMS Datenunabhängigkeit (Geisler, 2009, S. 50-53).

Datenträgerverwaltung

Das einzige Programm, das auf die physikalisch gespeicherten Daten der Datenbank zugreifen kann, ist das DBMS. Es erzeugt komplexe Strukturen auf der Festplatte, um Daten effizient und leistungsfähig zu speichern (Geisler, 2009, S. 50-53).

Metadatenverwaltung

Als letzte und einer der wichtigsten Aufgaben hat das DBMS die Funktion, Metadaten zu verwalten. Über eigens vorgesehene Funktionen werden diese im so genannten Katalog gespeichert (Geisler, 2009, S. 50-53).

3.5 *Datenbankmodelle*

Datenbankmodelle sind eine abstrahierte Darstellung der Beziehungen zwischen Daten zueinander. Es gibt zwei Modelle, konzeptionelle und implementative, die wiederum in kleinere Einheiten unterteilt werden. Konzeptionelle Modelle beschreiben die logische Struktur der Daten und damit, was in die Datenbank aufgenommen werden soll. Implementative Modelle hingegen schildern, wie die Daten gespeichert werden und wie die Beziehungen der erzeugten Strukturen zueinanderstehen.

3.5.1 Implementative Datenbankmodelle

Hierarchische-Datenbanken

Hierarchische Datenbanken sind die ältesten Datenbankmodelle und zeigen Daten in einer Baumstruktur mit Rangordnungen. Es wird mithilfe eines so genannten Zeigers gearbeitet. Der Inhalt eines Speicherplatzes verweist wiederum auf einen anderen Speicherplatz. Somit wird eine Abfrage intern so oft ausgeführt, bis die gewünschte Datei erreicht ist (Geisler, 2009, S. 53 – 74).

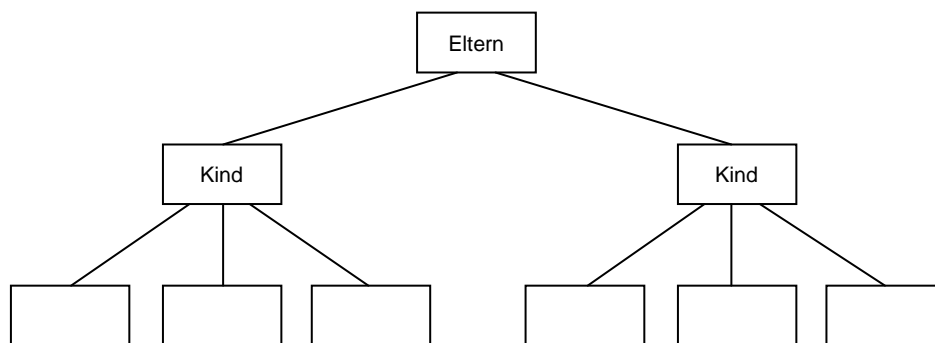


Abb. 3-5: Hierarchische-Datenbank

Netzwerk-Datenbanken

Netzwerk Datenbanken sind eine Weiterentwicklung der hierarchischen Datenbanken. Die Baumstruktur ähnelt der oben beschriebenen, es sind aber pro Kindknoten mehrere Elternknoten möglich (Geisler, 2009, S. 53 – 74).

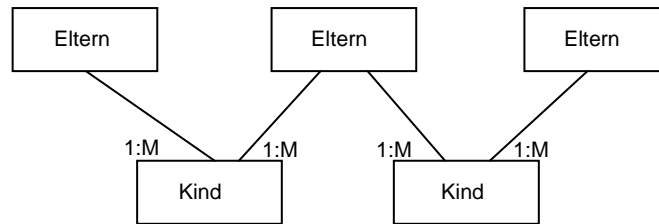


Abb. 3-6: Netzwerk-Datenbank

Relationale-Datenbanken

Die Art der relationalen Datenbank ist der zurzeit am meisten verwendete Typ. Einer der größten Vorteile besteht in der kompletten Kapselung der physikalischen Datenspeicherung. Ein Anwender hat sich nur mit der logischen Struktur der Daten zu befassen. Die Speicherung hingegen muss nicht beachtet werden. Die relationale Datenbank baut auf Tabellen auf, die aus zweidimensionalen Matrizen, sprich Zeilen und Spalten, bestehen. Beziehungen zwischen einzelnen Tabellen werden über Primär- oder Fremdschlüsselbeziehungen vorgenommen (Geisler, 2009, S. 53 – 74).

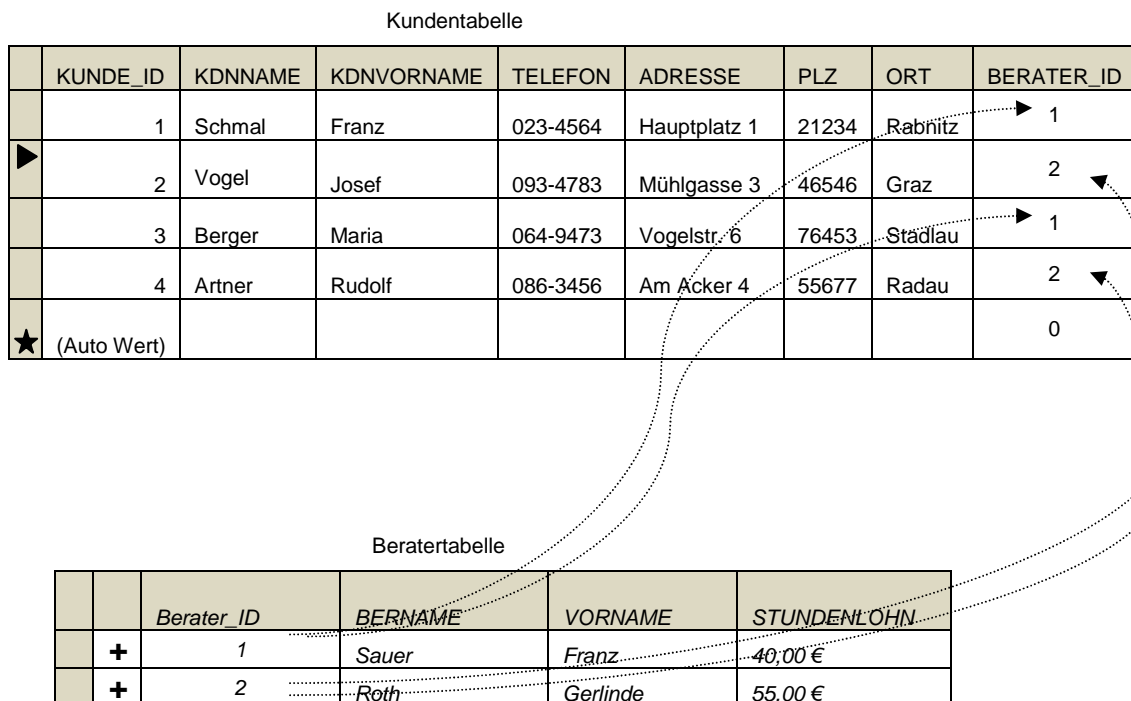


Abb. 3-7: Verbindung zwischen zwei Tabellen über Primär- /oder Fremdschlüssel (vgl. Geisler, 2009, S. 63)

3.5.2 Konzeptionelle Datenbankmodelle

Entity-Relationship-Datenbankmodelle (ER-Modelle)

Mithilfe des Entity-Relationship Modells werden Strukturen der zu erfassenden Daten erarbeitet. Als zweite Komponente werden Entitäten (ein Objekt in der realen Welt, das in der Datenbank verwaltet wird) und Beziehungen, mit Zuhilfenahme von Verben, von Daten dargestellt (Geisler, 2009, S. 53 – 74).

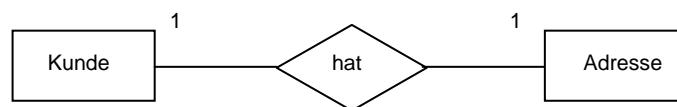


Abb. 3-8: 1:1 Beziehung zwischen Kunde und Adresse (vgl. Geisler, 2009, S. 69)

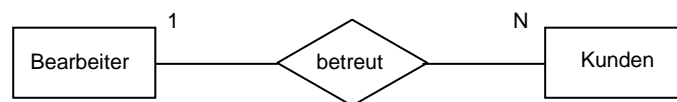


Abb. 3-9: 1:N Beziehung zwischen Bearbeiter und mehreren Kunden (vgl. Geisler, 2009, S. 69)

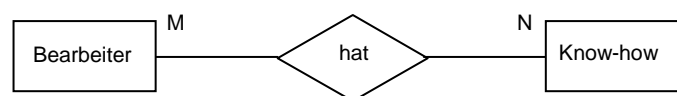


Abb. 3-10: M:N Beziehung zwischen Bearbeiter und Know-how (vgl. Geisler, 2009, S. 69)

Objektorientierte Datenbanken

In objektorientierten Datenbanken wird zur Oberflächenprogrammierung der Datenbankanwendung eine objektorientierte Programmiersprache verwendet. Im Hintergrund arbeitet eine relationale Datenbank. Es werden so genannte Klassen gebildet, die einmal angelegt, weitervererbt werden (Geisler, 2009, S. 53 – 74).

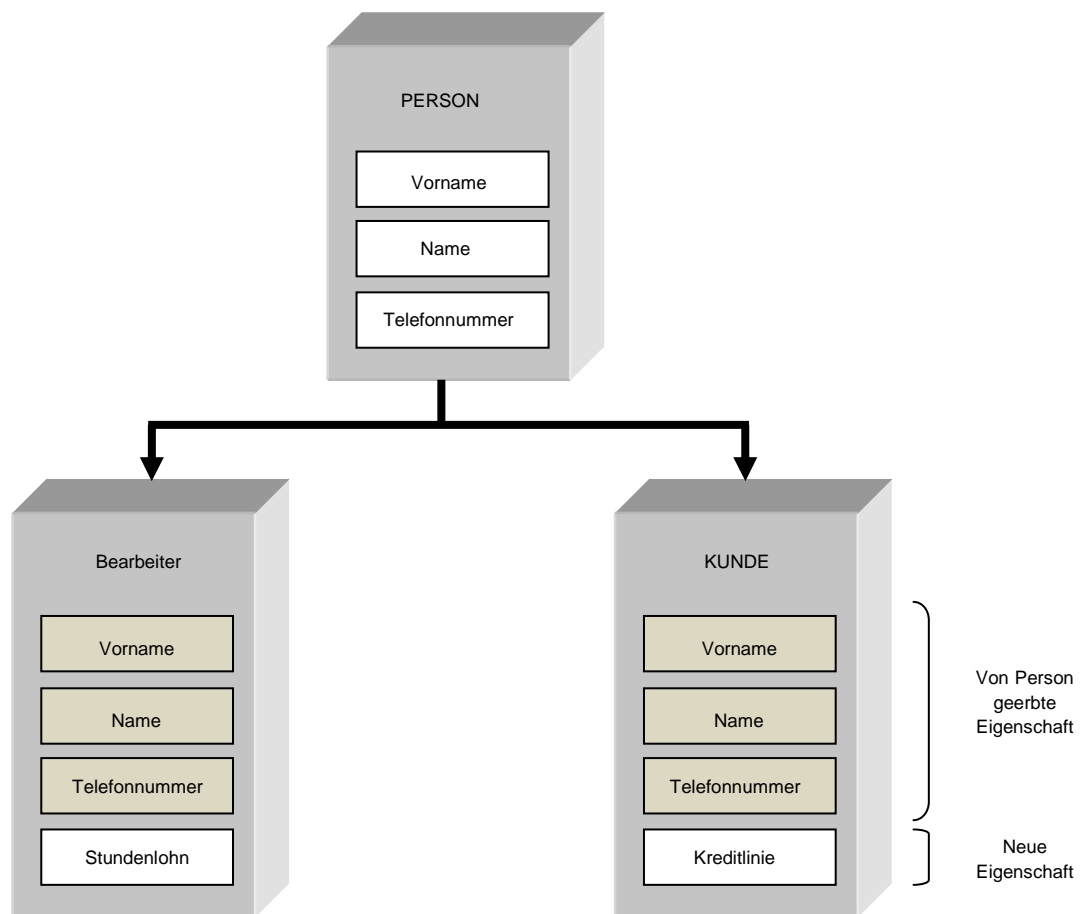


Abb. 3-11: Eine einfache Klassenhierarchie (vgl. Geisler, 2009, S. 74)

4 LabVIEW™

LabVIEW™, von National Instruments, erschien erstmals 1986 als eine Version für Macintosh-Computer. Heute gibt es diese Entwicklungsumgebung zusätzlich für Windows, Linux und Solaris (Wikipedia, 2010).

LabVIEW™ dient als Grundlage für die programmierte Reparaturdatenbank. Daher werden in diesem Kapitel grundlegende Komponenten und Werkzeuge, die zur Programmierung dienen, beschrieben.

In LabVIEW™ werden Symbole anstatt Textzeilen verwendet, um Anwendungen zu erstellen. Es handelt sich hier also um eine grafische Programmiersprache. Anstatt von befehlsorientierter, wie beim textbasierenden Programmieren, spricht man hier von datenflussorientierter Abfolge. Die zeitliche Ausführung von virtuellen Instrumenten (VIs) und Funktionen geschieht in Abhängigkeit vom Datenfluss und deren Knoten (in textbasierten Programmiersprachen sind das Anweisungen, Operatoren, Funktionen und Subroutinen) im Blockdiagramm. Mithilfe dieser VIs werden Geräte, echte Messinstrumente wie z. B. Multimeter und Oszilloskope, nachgebildet. In LabVIEW™ wird als erster Schritt eine Bedienoberfläche mithilfe von Werkzeugen und Objekten erstellt. Diese Oberfläche heißt Frontpanel. Bedien- und Anzeigeelemente, z. B. Eingabeelemente wie Drehschalter, Druckschalter oder Drehregler, bzw. Ausgabeelemente, z. B. Graphen oder LEDs, dienen zur Erstellung der Oberfläche. Ein neuer Programmcode wird durch Hinzufügen von grafischen Darstellungen diverser Funktionen geschrieben. Man bezeichnet diesen grafischen Quellcode als G- oder Blockdiagrammcode. Dieser befindet sich im Blockdiagramm und hat im Einzelnen Ähnlichkeit mit einem Flussdiagramm (2005 National Instruments Corporation, S. 1-1 – 2-4).

„LabVIEW™ ermöglicht die Kommunikation mit unterschiedlichster Hardware wie etwa Datenerfassungskarten, Bildverarbeitungs- und Motorensteuerungsmodulen sowie GPIB-, PXI-, VXI- und seriellen Geräten (RS-232 und RS-485)“ (2003-2006 National Instruments Corporation, S. 1-1).

4.1 Komponenten eines VIs

Ein VI benutzt Funktionen, die die Eingaben über die Benutzeroberfläche oder Informationen aus anderen Quellen z. B. Messergebnisse umsetzen. Diese Daten können in Form von Anzeigen ausgegeben oder in Dateien gespeichert werden.

Ein VI besteht grundsätzlich aus 3 Komponenten. Das Frontpanel stellt die Benutzeroberfläche dar. Das Blockdiagramm beinhaltet den G-Code, der die Funktionen des VIs vorgibt (National Instruments Corporation, 2006, S. 2-1 – 3-4).

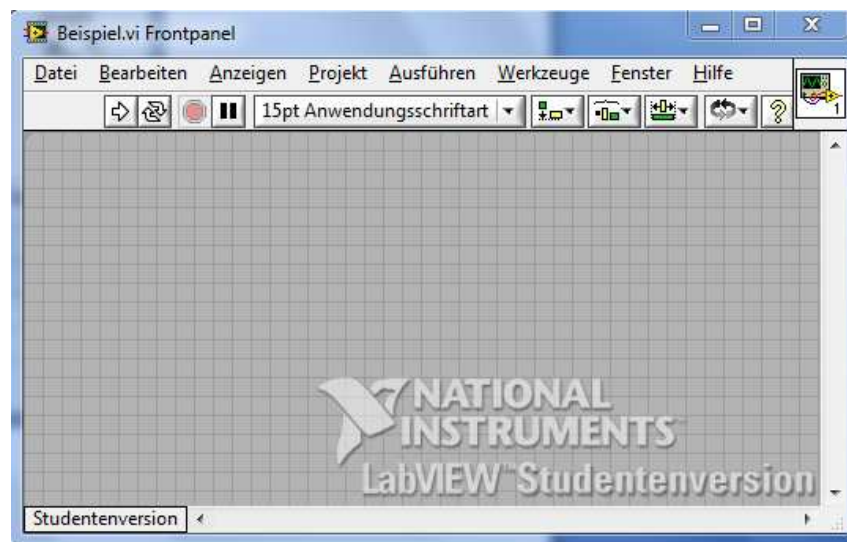


Abb. 4-1: Frontpanel

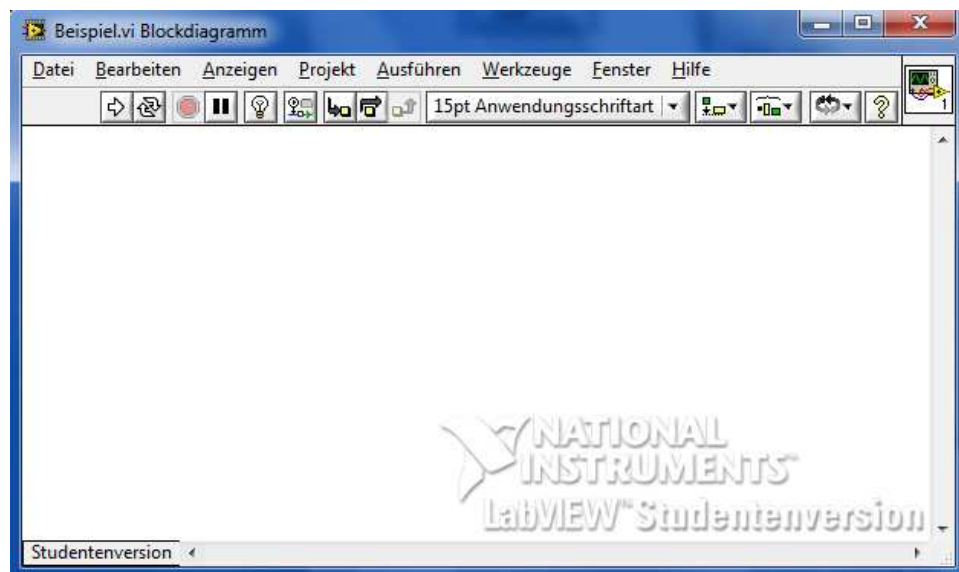


Abb. 4-2: Blockdiagramm

Symbol- und Anschlussfelder werden benötigt, um ein VI als Funktionsprototypen zu kennzeichnen, damit es öfter verwendet werden kann (siehe Kapitel 4.1 „Unterprogramme“). Nach erfolgter Programmierung wird dem VI ein Symbol zugeordnet und das dazugehörige Anschlussfeld erstellt. Diese werden gebraucht, um bei Verwendung als Sub-VI - in textbasierten Programmiersprachen wäre hier von einem Unterprogramm die Rede - definierte Ein- und Ausgänge zu erhalten, die mit dem Haupt-VI verbunden werden (National Instruments Corporation, 2006, S. 2-1 – 3-4).

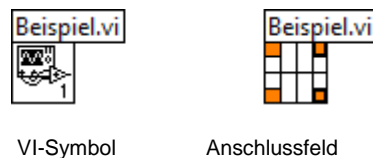


Abb. 4-3: Symbol- und Anschlussfeld

Unterprogramme (Sub-VIs)

In allen Programmiersprachen findet man Unterprogramme im Programmcode. Sie werden aus folgenden Gründen verwendet:

- Ein häufig benutzter Code muss nicht immer neu programmiert werden.
- Der Code wird nur 1 Mal gespeichert.
- Speziell in LabVIEW™ kann mit Sub-VIs die Anzahl der Symbole auf dem Bildschirm reduziert werden. Somit bleibt das Programm übersichtlich (Georgi & Metin, 2009, S. 93).

Abbildung 4-3 zeigt ein Sub-VI-Symbol mit dazugehörigem Anschlussfeld.

Im Blockdiagramm werden die Symbole des Frontpanels als Datentyp-Symbol (siehe Abb. 4-4 (4)) oder als Bild (siehe Abb. 4-4 (3)) dargestellt.

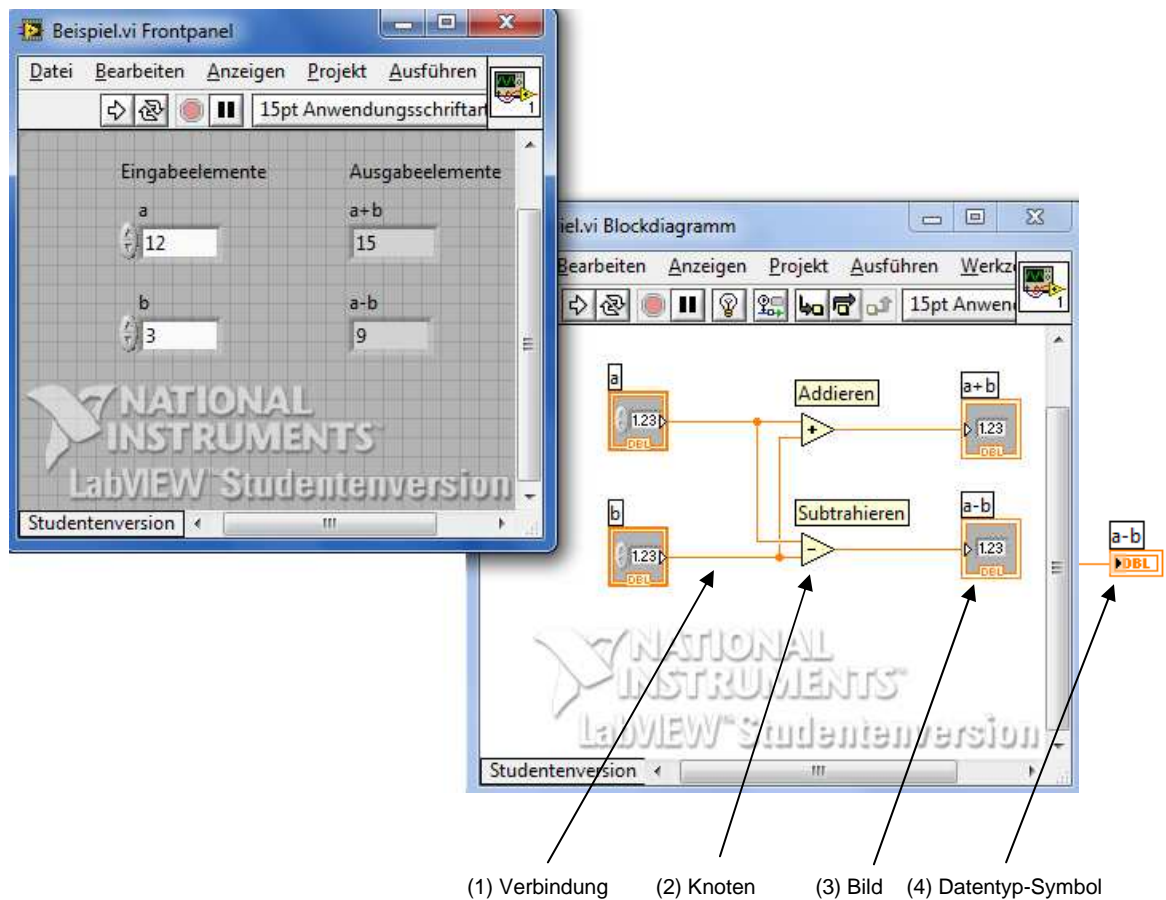


Abb. 4-4: Beispiel VI mit Frontpanel und dazugehörigem Blockdiagramm

Objekte mit Ein- und/oder Ausgängen bezeichnet man als Knoten (siehe Abb. 4-4 (2)). Sie sind beispielsweise Operatoren, Anweisungen, Funktionen oder Subroutinen in textbasierten Programmiersprachen. In der oben gezeigten Abbildung sind dies z. B. Additions- und Subtraktionsfunktionen. Um Daten zwischen den einzelnen Blockdiagrammobjekten auszutauschen, müssen Verbindungen (siehe Abb. 4-4 (1)) hergestellt werden. Dies wird mithilfe der oben abgebildeten Linien im Blockdiagramm realisiert. Jede Verbindung hat eine Datenquelle (Ausgang) und ein oder mehrere Datensenken (Eingänge). Farben, Formate und Linienstärke variieren je nach Datentyp (Abb. 4-5). Schleifen- und Case-Anweisungen werden als Strukturen (Abb. 4-10) dargestellt. Einzelne Abschnitte werden so wiederholt, bedingungsabhängig gemacht oder in einer bestimmten

Reihenfolge abgearbeitet. Je nach Datentyp werden die Symbole in verschiedenen Farben dargestellt (National Instruments Corporation, 2006, S. 2-1 – 3-4).

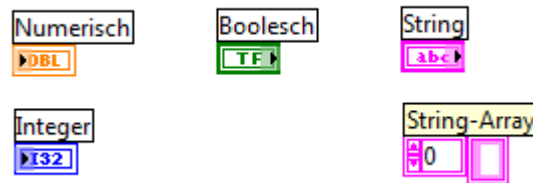


Abb. 4-5: Farbcodierung verschiedener Datentypen

4.2 Verwendete Programmierwerkzeuge

4.2.1 Palette mit Bedienelementen

Die Bedien- und Anzeigeelemente-Palette wird zur Erstellung des Frontpanels benötigt. Die Symbole der Elementepalette stellen die Bedien- und Anzeigeelemente dar, welche mittels Mausbedienung am Frontpanel angeordnet werden können. Alle hinzugefügten Elemente am Frontpanel werden automatisch als Bildsymbol oder Datensymbol (Abb. 4-4 (3), (4)) in das Blockdiagramm-Fenster (Abb. 4-2) eingefügt, wo sie mit dem ausführbaren Blockdiagramm verbunden werden müssen. Bedienelemente werden dazu benötigt, um den Programmablauf durch den Anwender zu beeinflussen. Anzeigeelemente dienen zur Darstellung von Daten. Diese Elemente stellen den sichtbaren Teil des VIs dar.

Aufgrund der großen Anzahl von Elementen ist die Palette in mehrere Unterpaletten gegliedert, damit der Programmentwickler einen besseren Überblick erhält (National Instruments Corporation, 2006, S. 2-1 – 3-4).

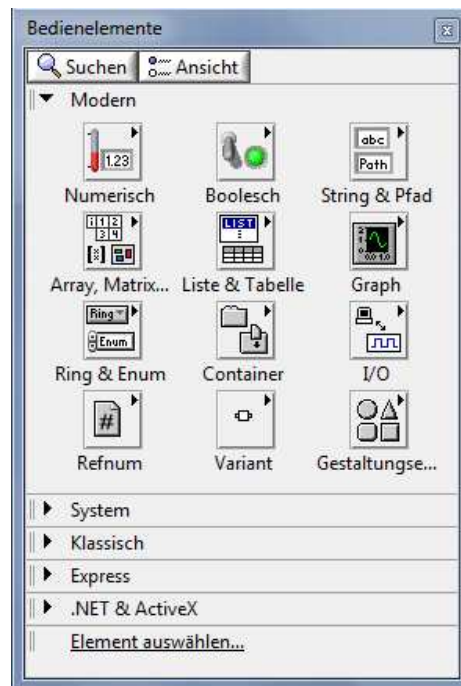


Abb. 4-6: Bedien- und Anzeigeelemente-Palette

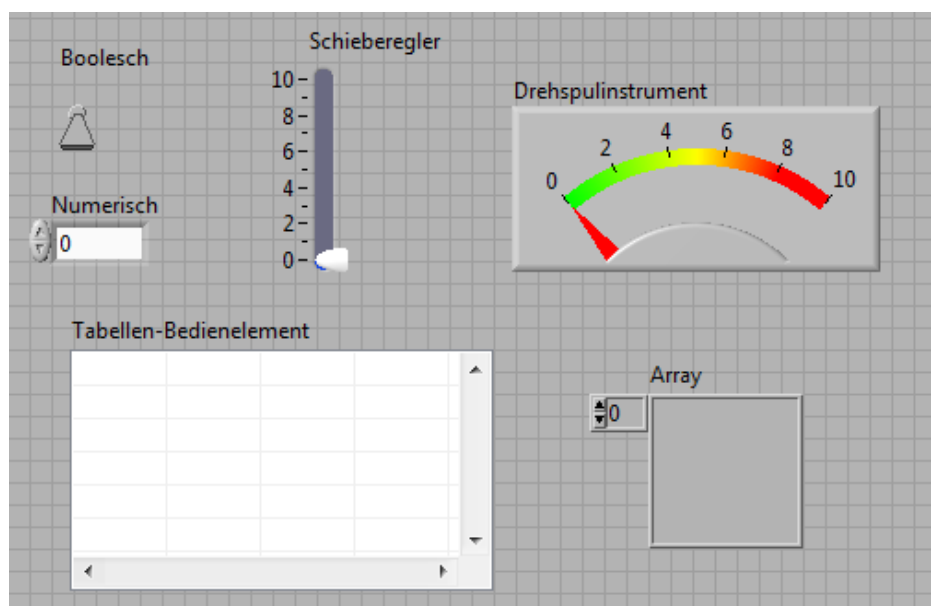


Abb. 4-7: Beispiele von möglichen Anzeige- und Bedienelementen

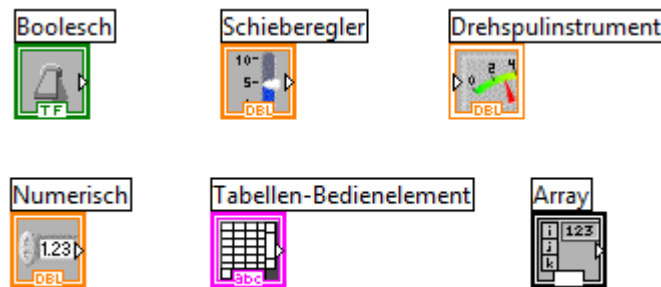


Abb. 4-8: Anzeige- und Bedienelemente (zu Abb. 4-7) im Blockdiagramm

4.2.2 Funktionenpalette

Die Funktionenpalette beinhaltet Symbole, welche zur Erstellung des ausführbaren Blockdiagramms verwendet werden können. Mithilfe dieser Symbole können zahlreiche vordefinierte Funktionen und Strukturelemente mittels Mausbedienung in das Blockdiagramm eingefügt werden. Um das VI ausführen zu können, müssen die einzelnen Symbole zu einem Blockdiagramm verbunden oder auch „verdrahtet“ werden. Das Blockschaltdiagramm stellt das eigentliche Programm dar (National Instruments Corporation, 2006, S. 2-1 – 3-4).

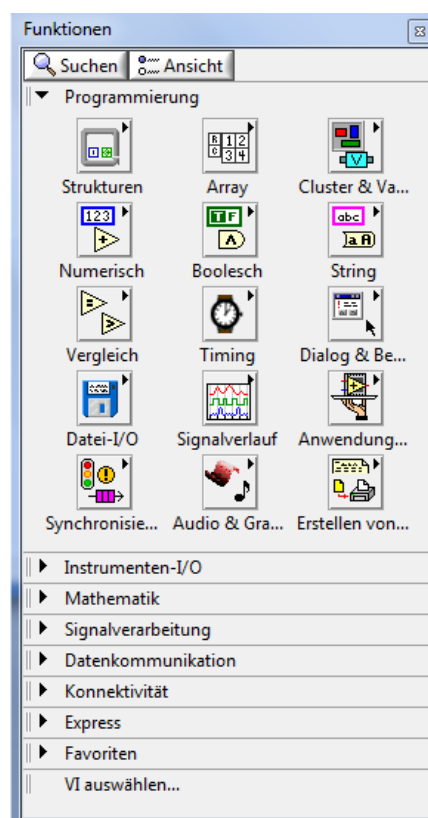


Abb. 4-9: Funktionenpalette

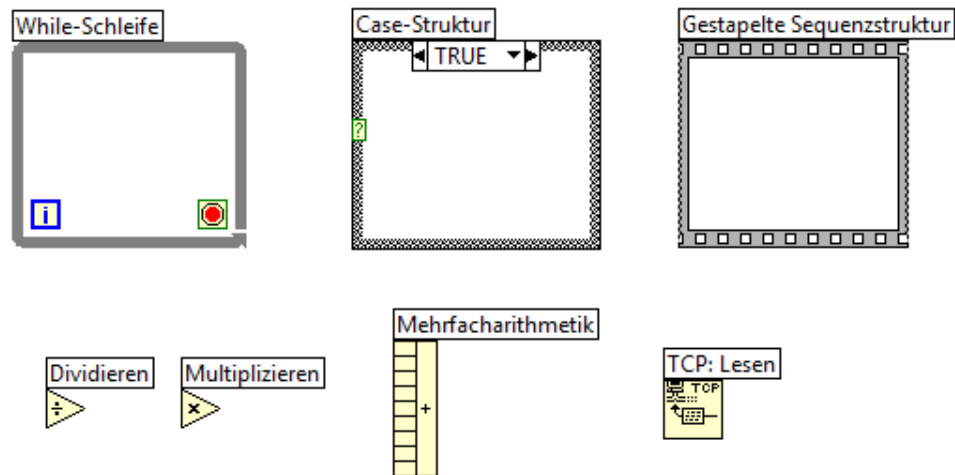


Abb. 4-10: Ein kleiner Teil der Strukturelemente und Funktionen

4.2.3 Werkzeugpalette

Die Palette mit den Werkzeugen steht auf dem Frontpanel und dem Blockdiagramm zu Verfügung. Je nach Auswahl wird der Mauszeiger zu einem bestimmten Werkzeug (National Instruments Corporation, 2006, S. 2-1 – 3-4).

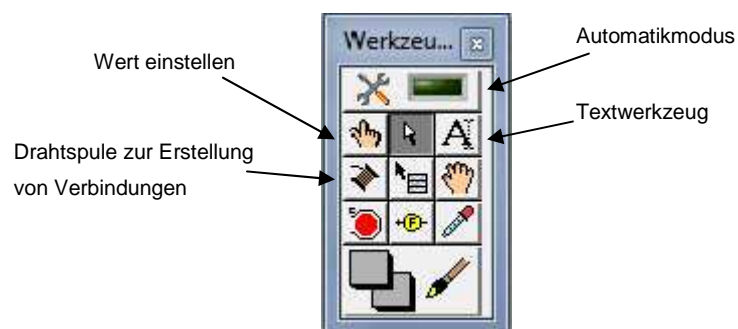


Abb. 4-11: Werkzeugpalette mit Beispielen

5 Anforderungen an die Reparaturdatenbank

Mit der Datenbank soll die Verwaltung von Geräten, welche als Reklamation oder kostenpflichtige Reparatur in die Fertigung gelangen, durchgeführt werden. Da entweder ein Produkt oder mehrere, in einem Auftrag zusammengefasst, in die Reparaturabteilung geschickt werden, soll eine Erfassung über die Seriennummer (Abb. 5-1(1)) und einer Auftragsnummer erfolgen. Die Seriennummer ist eine 10-stellige Zahl, die sich aus dem Produktionsjahr und einer laufenden Nummer zusammensetzt. Die Auftragsnummer - vom Warenwirtschaftsprogramm vergeben - ist 10-stellig und setzt sich aus dem Buchstaben S (für Siegendorf), dem Kalenderjahr und einer laufenden Nummer zusammen. Alle Geräte, die in die Reparaturabteilung gelangen, sollen sofort in der Datenbank registriert werden. Bei Bearbeitung der Reparatur sollen Daten abgerufen, ergänzt und letztendlich als fertige Reparatur abgespeichert werden. Zu jedem fehlerhaften Gerät sollen folgende Daten verarbeitet werden:

- Seriennummer und dazugehörige Reparaturauftragsnummer
- Eingangsdatum und Abschluss - / Enddatum der Bearbeitung
- Produkttyp über Angabe der Gerätenummer laut Typenschild
- Fehlerbeschreibung des Kunden
- Durchgeführte Tests und eventuelle Softwareupdates
- Aufgewendete Bearbeitungszeit
- Name des Technikers über ein firmeninternes Kurzzeichen
- Endstatus des Gerätes (repariert, irreparabel, Ersatzgerät)
- Fehlerart/Fehlerursache, defekter Bauteil
- Wiederholter Fehlerfall des Gerätes

Serien- und Gerätenummer werden anhand des Typenschildes abgelesen.



Abb. 5-2: Typenschild mit Barcode

6 Struktur des Programms

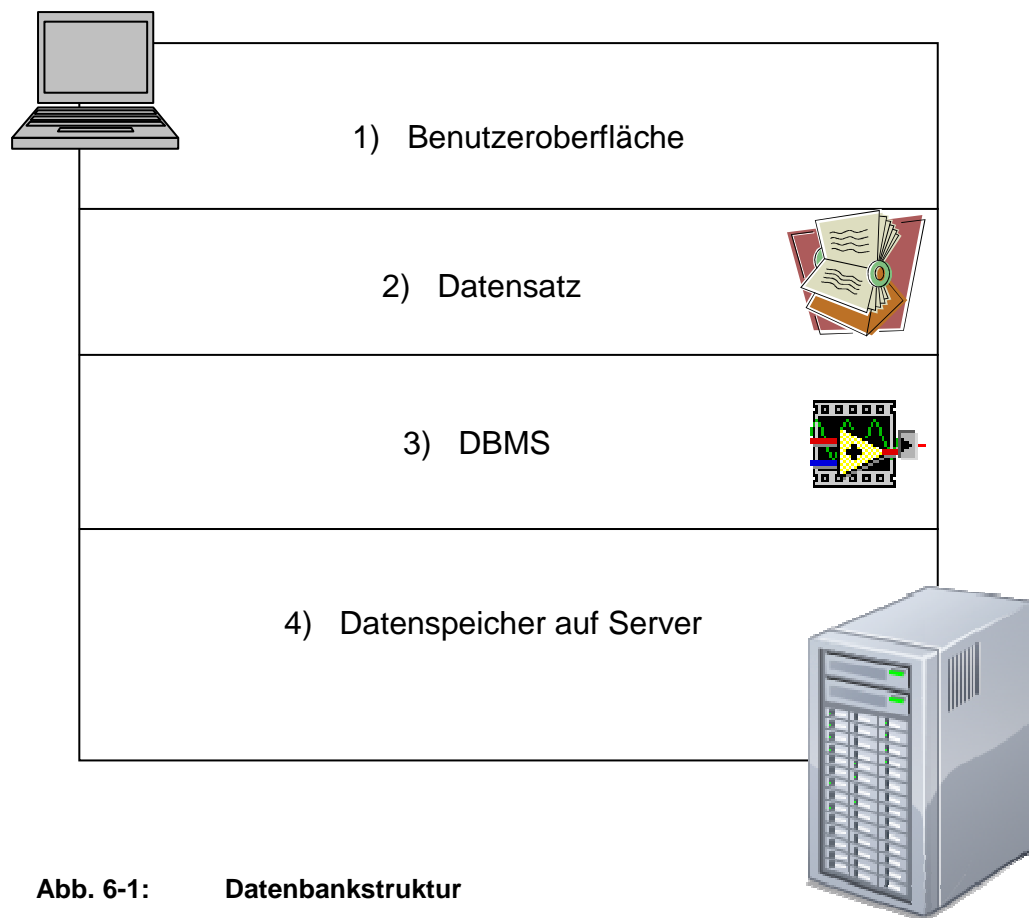


Abb. 6-1: Datenbankstruktur

- 1) Die Benutzeroberfläche ermöglicht Eingabe, Laden, Speichern, Löschen und Ansehen von Datensätzen.
- 2) Ein Datensatz beinhaltet sowohl die Nutzdaten als auch Metadaten. Nach Bearbeitung wird dieser in Form einer flachen Datei als Textfile abgelegt.
- 3) Das DBMS verwaltet die gesamten Daten. Bei Bedarf werden die erforderlichen Daten geladen oder gespeichert.
- 4) Der Datenspeicher auf dem Server beinhaltet alle Daten und sämtliche Hilfsdateien. Es wird ein tägliches Backup der Serverdaten auf Magnetbändern durchgeführt.

Bei der gegenständlichen Datenbank handelt es sich um eine Einzelbenutzer-Desktop-Datenbank.

6.1 VI-Hierarchie der Reparaturdatenbank

Die Abbildung 6-2 zeigt die Hierarchie des gesamten Datenbankprogramms mit dem Hauptprogramm und den Sub-VIs.

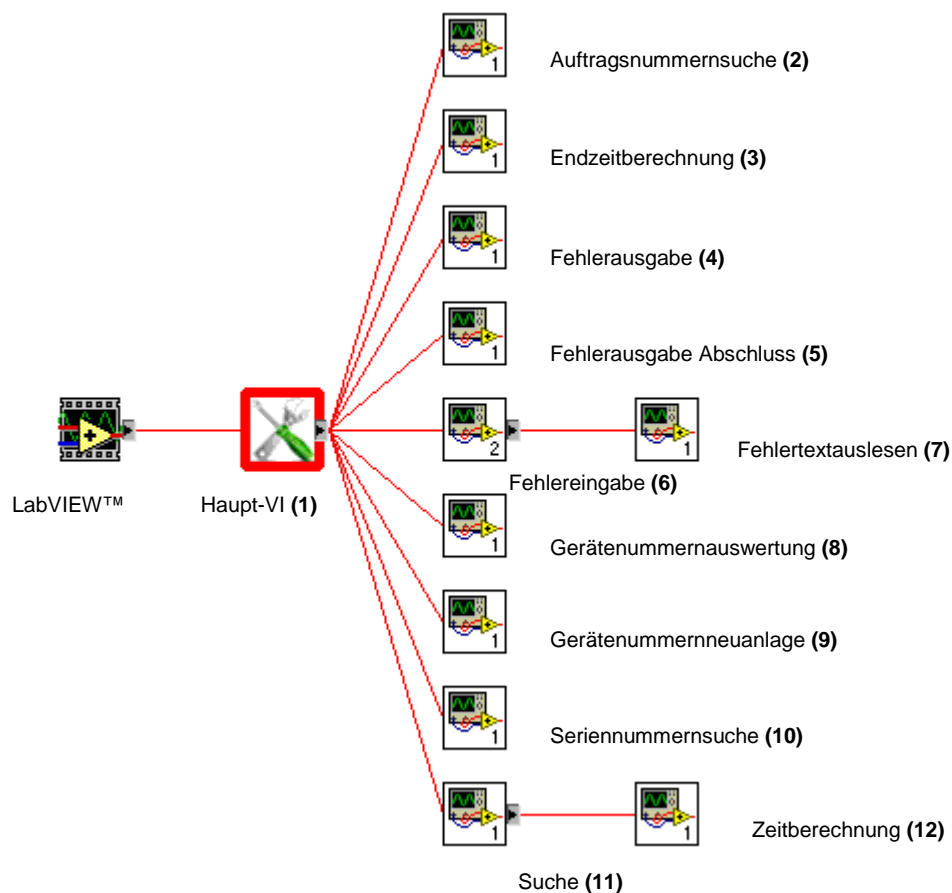


Abb. 6-2: VI-Hierarchie

Das Hauptprogramm (siehe Abb. 6-2 (Haupt-VI)) beinhaltet die Benutzeroberfläche, auf der alle Eingabefelder sowie die Taster und Schalter, abgebildet sind, die zur Interaktion mit der Datenbank nötig sind. Weiters befindet sich der aktuell dargestellte Datensatz im Hauptprogramm. Die Funktionen des DBMS werden teilweise im Hauptprogramm und großteils in den darunterliegenden Sub-VIs ausgeführt.

6.2 *Kurzbeschreibung der VIs und Sub-VIs*

Die folgende Tabelle gibt eine kurze Beschreibung der einzelnen VIs und Sub-VIs wieder. Die VI-Nummern sind von der Abb. 6-2 übernommen worden. Neben der Funktion wird über die Spalte "Kapitel" auf das Kapitel, in dem eine nähere Beschreibung der Programmteile erfolgt, hingewiesen.

VI Nr.	Kurzbeschreibung / Funktion	Kapitel
1	Hauptprogramm / Benutzeroberfläche Ein- und Ausgabe	7.1
2	Durchsucht alle Datensätze nach einer bestimmten Auftragsnummer.	7.2.4
3	Berechnet die verbleibende Reparaturzeit.	7.2.2
4	Generiert eine Fehlermeldung bei Falscheingabe bei der Speicherung der Ersteingabe.	7.1.1
5	Generiert eine Fehlermeldung bei Falscheingabe beim abschließenden Speichern des Datensatzes.	7.3.3
6	Schreibt die Anzeige der Fehlercodes samt Klartext und Bauteilbenennung von der Benutzeroberfläche in einen String, der in den Datensatz eingefügt wird.	7.3.3
7	Sucht den Fehlercode in einer Datei und gibt den dazugehörigen Klartext aus.	7.3.3
8	Sucht die Gerätenummer in einer Datei und gibt den dazugehörigen Gerätenamen aus.	7.1.1
9	Dient zur Neuanlage einer Gerätenummer mit dazugehörigem Gerätenamen.	7.4.1
10	Durchsucht alle Datensätze nach einer bestimmten Seriennummer.	7.2.2
11	Lädt den Inhalt der mit Serien- oder Auftragsnummernsuche gefunden Datensätze je nach Auswahl in die Benutzeroberfläche.	7.2.2
12	Wandelt den gespeicherten Zeitstring eines Datensatzes in das Anzeigeformat der Benutzeroberfläche um.	7.2.2

Tabelle 6-1: Kurzbeschreibung VIs und Sub-VIs

7 Reparaturdatenbank

Die Beschreibung der Reparaturdatenbank erfolgt chronologisch zu einer durchgeführten Reparatur; zuerst aus Sicht des Benutzers und jeweils anschließend als Programmablauf mit benutzten Sub-VIs.

7.1 *Ersteingabe neuer Reparaturdaten*

Zunächst wird das Programm mit einem Doppelklick auf den Icon (Abb.7-1) gestartet. Die Eingabemaske wird geladen und auf dem Monitor angezeigt (Abb.7-2).

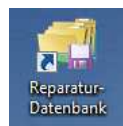


Abb. 7-1: Desktop Icon

The screenshot shows the 'Reparaturdatenbank' input mask (Frontpanel.vi) with the following elements:

- Buttons:** 'Neue Eingabe', 'SPEICHERN', 'Reset Eingabefenster', 'Datensatz löschen', 'Gerätenummer Neu', 'Suche Seriennummer', 'Suche Auftrag', 'Reparatur abschließen', 'Reparatur abgeschlossen', 'voriger Datensatz', 'nächster Datensatz', 'Programm beenden'.
- Input Fields:** 'Seriennummer (10-stellig)', 'Auftragsnummer (10-stellig)', 'Gerätenummer', 'Datum Eingang' (DD.MM.YYYY), 'Reparatur Zeit' (h), 'Bearbeiter', 'Repariert' (radio buttons), 'Irreparabel' (radio button), 'Ersatzgerät' (radio button), 'Datum Ende' (DD.MM.YYYY).
- Text Labels:** 'Dieses Gerät war bereits', 'mal zur Reparatur!', 'Gerät Nr.', 'von', 'Es verbleiben noch', 'Tage zur Fertigstellung!', 'Fehlerbeschreibung vom Kunden:', 'Tätigkeiten / Bemerkung:'.
- Table:** A table with 3 columns: 'Fehler Nr.', 'Fehlertext', and 'Bauteilbenennung laut Bestückungsplan'.
- Logo:** 'NATIONAL INSTRUMENTS LabVIEW Studentenversion'.

Abb. 7-2: Eingabemaske

Die Reihenfolge der auszufüllenden Felder erfolgt von links oben nach rechts unten. So kann sichergestellt werden, dass auch alle Felder ausgefüllt sind. Die Seriennummer kann entweder manuell eingegeben oder mit einem Barcodescanner direkt vom Typenschild des Gerätes eingelesen werden. Danach erfolgt die Eingabe der Auftragsnummer, der Gerätenummer, des Datums und der Fehlerbeschreibung des Kunden. Die eingegebene Gerätenummer wird ausgewertet und die Typenbezeichnung im Feld darunter angezeigt (Abb. 7-3 (1)). Bei der Neuanlage einer Reparatur müssen alle grün blinkenden Felder (Abb. 7-3) korrekt ausgefüllt werden, da sonst keine Speicherung möglich ist. Wenn alle erforderlichen Angaben getätigt sind, erfolgt die Speicherung der Daten durch Anklicken des Tasters "Speichern" (Abb. 7-3 (2)).

Abb. 7-3: Felder für Neuanlage ausgefüllt

Sollte eine der Bedingungen der Eingabefelder nicht erfüllt sein, sprich Serien- und/oder Auftragsnummer zu kurz/lang, Gerätenummer, Fehlerbeschreibung vom Kunden oder Datum Eingang fehlen, erscheint ein Popup mit der Fehlermeldung "Bitte Eingaben überprüfen!" (Abb. 7-4).

Abb. 7-4: Fehlermeldung

Nach erfolgreicher Speicherung der Daten wird die Anzeige auf dem Bildschirm gelöscht. Nun ist die nächste Eingabe oder Abfrage möglich.

7.1.1 Programmablauf Neuanlage

Alle Zeichen, welche in den fünf Feldern der Bedienoberfläche eingegeben worden sind, werden als String zusammengefasst, wobei zwischen den Daten drei Schrägstriche “///” als Trennzeichen eingefügt werden. Dieser String wird im Ordner Datensätze mit der jeweiligen Seriennummer und Auftragsnummer als Namen in einem Textfile abgespeichert (Abb. 7-5). Über die Funktionen “Leerer String” und “String-Länge” werden die Eingaben der Felder kontrolliert (Abb. 7-6). Ein Ausschnitt der Funktion “Neuanlage“, die im Haupt-VI integriert wurde, ist in Abbildung 7-7 ersichtlich.

Dateiname: 2009000235_S201000001.txt

{
}
{
}

1 2

Dateiinhalte: 2009000235///S201000001///0404-1981GS2///22.01.2010 18:00:17///keine Funktion///

{
}
{
}
{
}
{
}

1 2 3 4 5

Abb. 7-5: Dateiname und Dateinhalt einer Neueingabe

Beschreibung der einzelnen Teile einer Datei von Abbildung 7-5.

1	Seriennummer des Gerätes
2	Auftragsnummer des Reparaturauftrages
3	Gerätenummer/Typ
4	Eingangsdatum/Erfassungsdatum der Reparatur
5	Fehlerbeschreibung des Kunden

Tabelle 7-1: Dateiinhalt einer Neuanlage

Abbildung 7-6 zeigt die zur Eingabekontrolle verwendeten Funktionen.

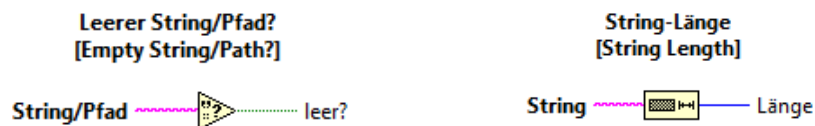


Abb. 7-6: Funktion leerer String

Funktion String-Länge

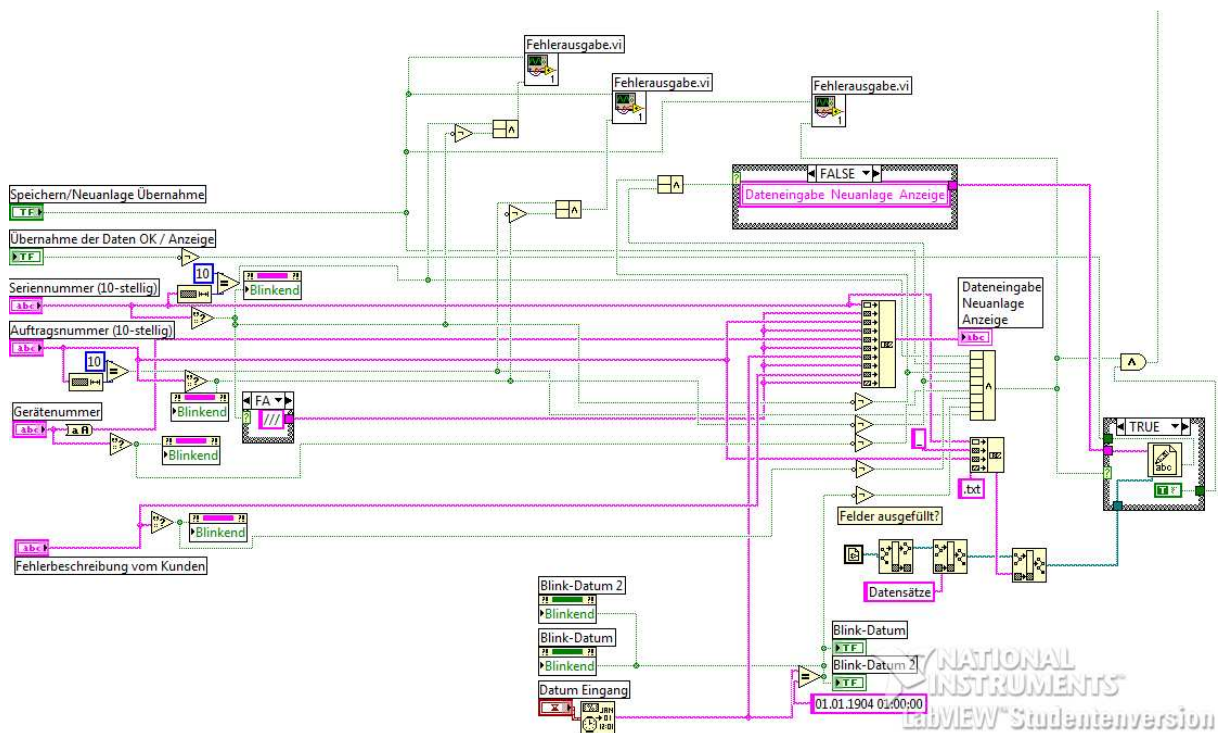


Abb. 7-7: Blockdiagramm Neuanlage (Ausschnitt)

Die Gerätenummernanzeige wird mithilfe eines Sub-VIs ausgeführt. Gerätenamen und dazugehörige Nummern sind in einem Textfile gespeichert. Bei Eingabe einer Gerätenummer in das Feld wird der Name im Textfile gesucht und anschließend ausgegeben (Abb. 7-8).

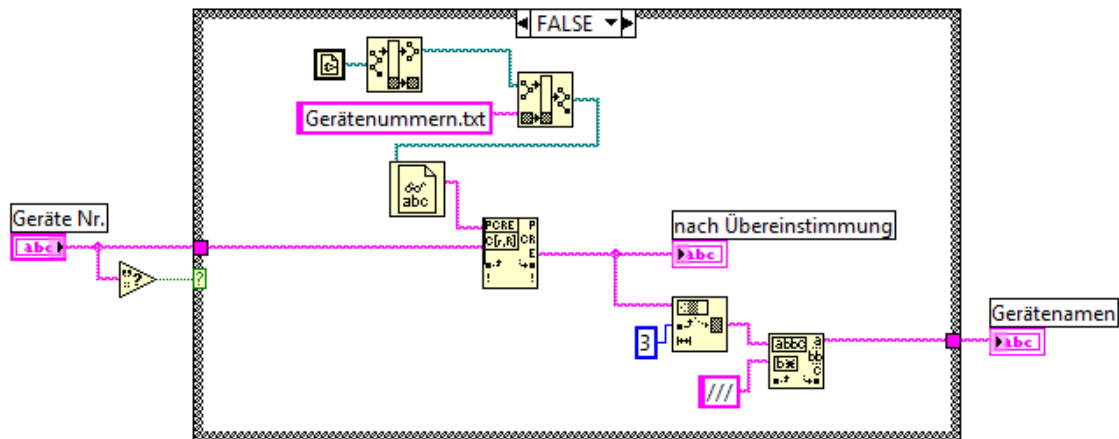


Abb. 7-8: Blockdiagramm Gerätenummernanzeige

Bei dem Versuch eine Neuanlage mit falsch oder nicht ausgefüllten Feldern zu speichern, generiert das in Abb. 7-9 gezeigte VI eine Fehlermeldung. Hierzu wird eine Funktion mit dem Namen "Dialogfeld" (Abb. 7-9 (1)) verwendet.

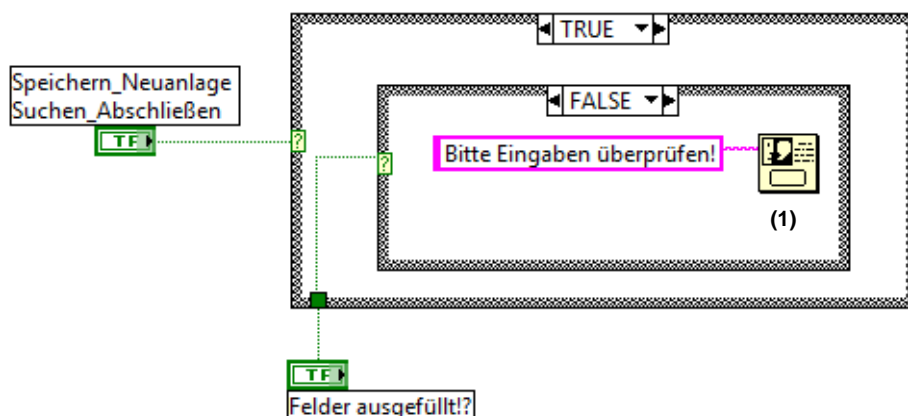


Abb. 7-9: Blockdiagramm Fehlerausgabe

7.2 Abfrage einer Ersteingabe

Wenn ein Gerät vom zuständigen Techniker zur Reparatur geholt wird, öffnet dieser das Datenbankprogramm und sucht anhand der Seriennummer oder Auftragsnummer die bei der Ersteingabe des Gerätes erfassten Daten.

7.2.1 Seriennummernsuche

In das Feld Seriennummer wird das gesuchte Gerät eingescannt oder manuell eingegeben. Durch Betätigen der Taste “Suche Seriennummer” (Abb. 7-10 (1)) erfolgt eine Abfrage und alle vorhandenen Daten werden in die Bedienmaske geladen. Zusätzlich wird der Eingangszeitstempel mit dem aktuellen Datum verglichen und die vorhandene Restzeit, die vorgegebene Durchlaufzeit beträgt maximal 10 Tage, der Reparatur wird in Form von Tagen angezeigt (Abb. 7-10 (2)). Sollte die gesuchte Seriennummer bereits zu einem früheren Zeitpunkt zur Reparatur im Werk gewesen sein, werden alle zugehörigen Datensätze ebenfalls geladen. Auf der Bedienoberfläche erfolgt ein Hinweis, dass dieses Gerät bereits einmal oder öfters repariert wurde (Abb. 7-10 (3)). Mit den Tasten “nächster Datensatz” (Abb. 7-10 (5)) und “voriger Datensatz” (Abb. 7-10 (4)) kann zwischen den geladenen Datensätzen umgeblättert werden.

Abb. 7-10: Seriennummernsuche

7.2.2 Programmablauf Seriennummernsuche

Das Starten der Suchfunktion ist nur dann möglich, wenn eine 10-stellige Seriennummer eingegeben wurde. Alle Dateinamen der vorhandenen Datensätze werden in eine Hilfsdatei namens "Files" geschrieben. Diese Datei wird durchsucht und alle Auftragsnummern, die zur gesuchten Seriennummer gefunden werden, in ein Array geschrieben. Die Anzahl der eingetragenen Auftragsnummern wird ausgegeben (Abb. 7-10 (3)).

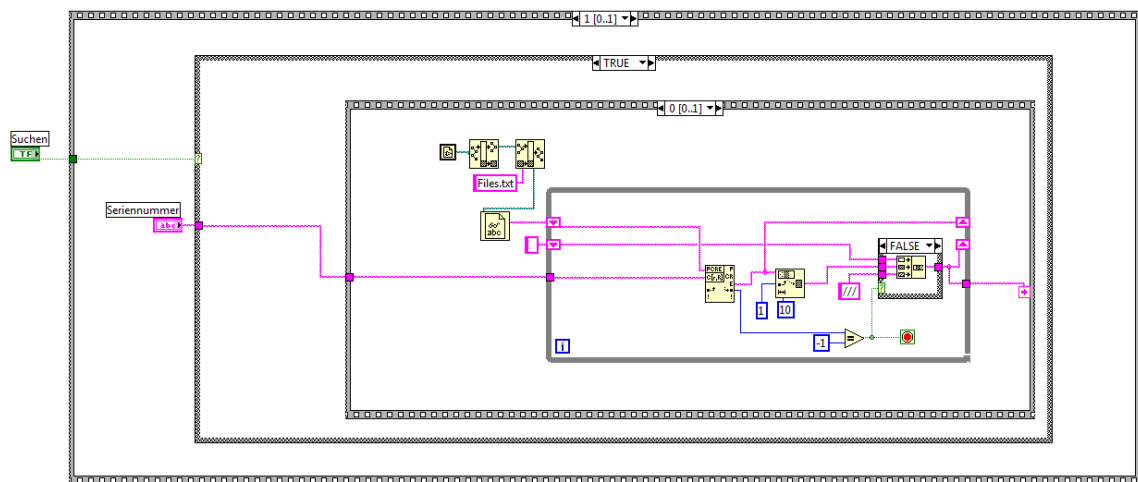


Abb. 7-11: Blockdiagramm Seriennummernsuche

Mithilfe des Sub-VIs "Suche" werden die Daten der aktuell angezeigten Seriennummer gesucht und der dazugehörige Datensatz auf dem Bedienfeld angezeigt. Mit einer Hilfsvariablen wird je nach Auswahl über die Schalter "nächster Datensatz" oder "voriger Datensatz" zwischen den einzelnen Datensätzen der Seriennummer geblättert.

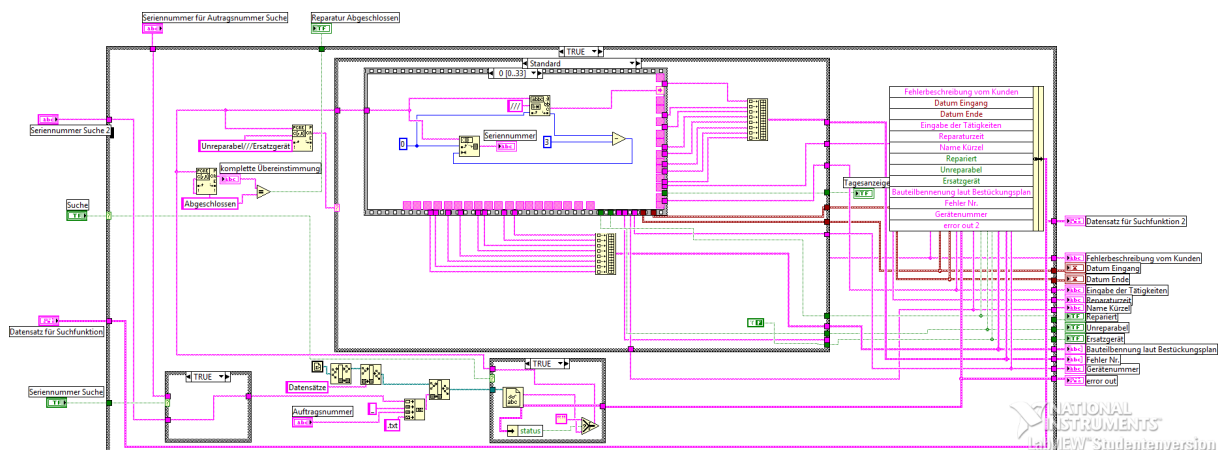


Abb. 7-12: Blockdiagramm Suchfunktion

Da der Zeitstempel in Form eines Strings gespeichert wird, muss bei einer Abfrage des Datensatzes mithilfe eines Sub-VIs der String in das Anzeigeformat der Datumsanzeige der Bedienoberfläche konvertiert werden (Abb. 7-13).

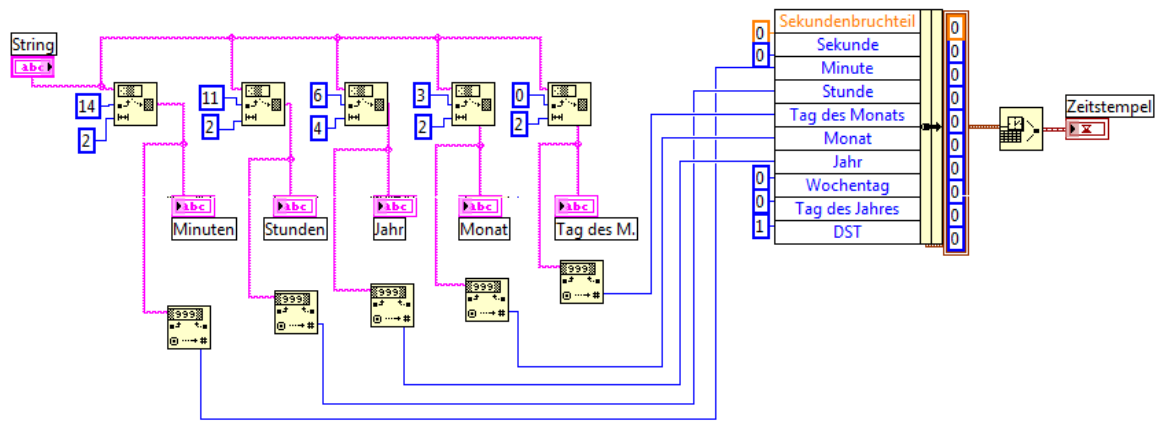


Abb. 7-13: Blockdiagramm String in Datumsanzeige

Das aktuelle Systemdatum und das Eingangsdatum der Reparatur werden subtrahiert und daraus die noch verbleibende Reparaturzeit errechnet (Abb. 7-14).

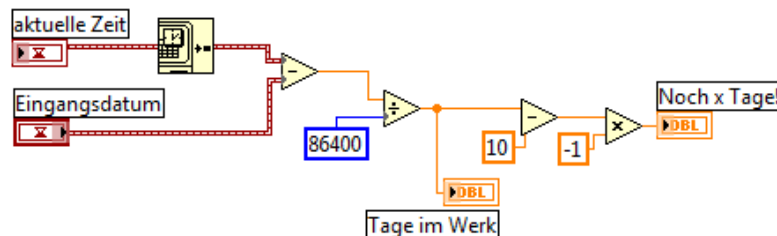


Abb. 7-14: Blockdiagramm Restzeitberechnung

7.2.3 Auftragsnummernsuche

Bei der Auftragsnummernsuche werden nach Eingabe einer 10-stelligen Auftragsnummer durch Betätigen der Taste "Suche Auftrag" (Abb. 7-15 (1)) alle Datensätze mit derselben Auftragsnummer angezeigt. Die Anzahl der gesamten Datensätze (Abb. 7-15 (2)) und der aktuell angezeigte Datensatz (Abb. 7-15 (3)) werden über eine Ausgabe am Bildschirm dargestellt. Mit den Tasten "nächster Datensatz" und "voriger Datensatz" kann wieder zwischen den geladenen

7.3 Abschluss des Bearbeitungsvorganges

7.3.1 Abschließende Eingabe

Nach erfolgter Suche eines Datensatzes wird dieser in der Regel weiterbearbeitet und abgeschlossen. Über die Eingabemaske können durchgeführte Arbeiten (Abb. 7-17 (1)), Bearbeitungsdauer (Abb. 7-17 (2)), Bearbeiter (Abb. 7-17 (3)), Endstatus (Abb. 7-17 (4)), Fehlernummer (Abb. 7-17 (5)), defekter Bauteil (Abb. 7-17 (7)) sowie das Enddatum (Abb. 7-17 (8)) der Reparatur erfasst werden. Der Fehlertext (Abb. 7-17 (6)), welcher den Defekt des zu reparierenden Gerätes beschreiben soll, wird aus einer Vorlage generiert. Anhand von vorgegebenen Fehlernummern (Tabelle 7-2) kann der Fehler eindeutig einer bestimmten Kategorie zugeordnet werden. Hierbei wird zwischen 3 Hauptgruppen unterschieden:

- bedrahtete Bauelemente
- SMD-R = Oberseite der Platine
- SMD-W = Unterseite der Platine

Durch die Angabe des defekten Bauteiles, laut Bestückungsplan (Abb. 7-18) des jeweiligen Gerätes, wird die Ursache der Reparatur in der Datenbank festgehalten. Es stehen insgesamt 8 Zeilen für die Fehlerangabe zur Verfügung (Abb. 7-17 (9)).

Reparaturdatenbank Version 1.3

Neue Eingabe **SPEICHERN** **Reset Eingabefenster** **Datensatz löschen** **Programm beenden**

Gerätenummer Neu

Suche Seriennummer: 2009004325 Suche Auftrag: S200900015 Gerätenummer: 0708-3101GS1 Datum Eingang: 11:32, 01.12.2009

Dieses Gerät war bereits 0 mal zur Reparatur! Gerät Nr. von Es verbleiben noch Tage zur Fertigstellung!

Fehlerbeschreibung vom Kunden:
SFP Modul wird nicht erkannt! Link geht nicht up, kein Pegel am Ausgangspegel!

Tätigkeiten / Bemerkung:
FT, ET, SW- Update
(1)

Fehler Nr.	Fehlertext	Bauteilbenennung laut Bestückungsplan
352	Mech. defekter Bauteil (SMD-W)	L 11
(5)	(6)	(7)
	(9)	

Reparatur Zeit: 1 h (2)
Bearbeiter: MRA
Repariert: (3)
Unreparabel:
Ersatzgerät:
(4)

Datum Ende: 11:35, 03.12.2009 (8)

voriger Datensatz Reparatur abschließen Reparatur abgeschlossen nächster Datensatz

NAT INST

Abb. 7-17: Vollständig ausgefüllte Eingabemaske

Jedes Bauteil eines Gerätes hat eine eindeutige Bezeichnung. So werden zum Beispiel Kondensatoren mit dem Buchstaben "C", Widerstände mit "R", Spulen mit "L", Dioden mit "D", Transistoren mit "T" und ICs mit "U" gekennzeichnet. Zusätzlich findet eine Nummerierung statt, damit jeder Bauteil einzigartig auf der Platine ist und so identifiziert werden kann z. B.: R 1, R 2, L 1, IC 4, D 9. Es ist darauf zu achten, dass diese Benennungen teilweise firmenspezifisch vergeben werden.

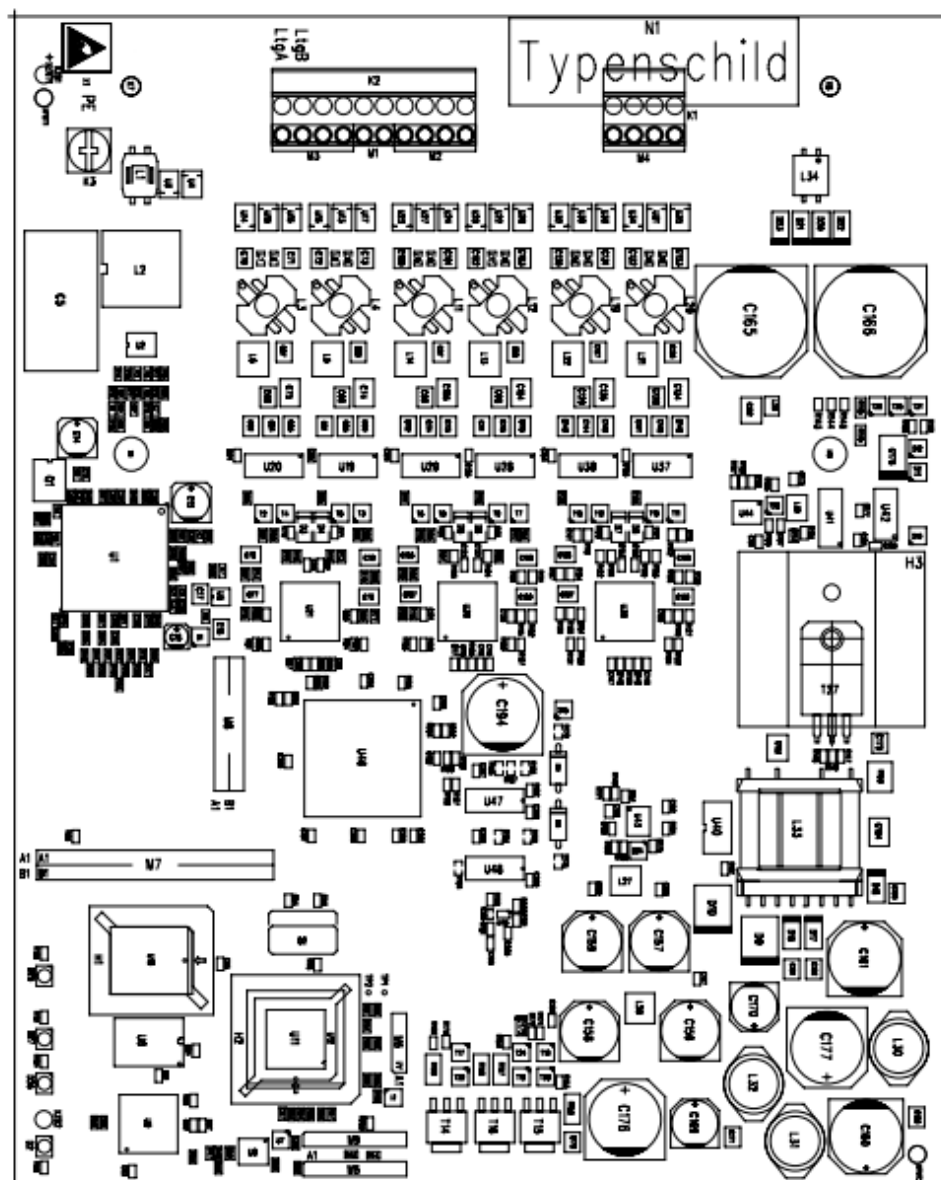


Abb. 7-18: Bestückungsplan eines Gerätes

SMD - W

SMD - W

SMD - W

Die verwendete Fehlerliste (Tabelle 7-2) wurde direkt von der Fertigung übernommen. Sie beinhaltet sowohl sämtliche Fehler, die im Zuge einer elektronischen Produktion auftreten können, als auch Fehler, die erst im Laufe des Betriebes eines Produktes auftreten. Bei der Handhabung der Fehlercodeliste hat sich gezeigt, dass im laufenden Reparaturbetrieb die gängigsten Fehlercodes aus dem Gedächtnis des Servicetechnikers eingetragen werden können. Dies ist der Grund, warum in der Praxis, aus eigener Erfahrung heraus, bei der Fehlereingabe mit den Fehlercodes und nicht mit Klartexteingabe gearbeitet wird. Die dadurch entstehende Zeitersparnis ist ebenso ein Vorteil dieser Vorgehensweise.

7.3.2 Speichern der Eingabe

Nach erfolgter Eingabe aller Daten kann nun die Reparaturhandlung abgeschlossen/gespeichert werden. Damit eine mangelhafte Eingabe der Daten vermieden werden kann, ist ein Speichern nur möglich, wenn alle notwendigen Felder ausgefüllt werden. Durch Drücken der Taste "Reparatur abgeschlossen" (Abb. 7-17 (1)) und Betätigen des Tasters "Reparatur abschließen" (Abb. 7-17 (2)) wird der Datensatz gespeichert und zusätzlich mit dem Status "Abgeschlossen" versehen. Sollten nicht alle Felder ausgefüllt sein, erscheint ein Popup mit der Fehlermeldung "Bitte Eingaben überprüfen!". Ebenso ist nur mit zu vorigem Schalten des Tasters "Reparatur abgeschlossen" eine Speicherung möglich. Ohne dessen Betätigung erscheint ein Fenster mit der Meldung "Bitte Schalter Reparatur abgeschlossen betätigen!" (Abb. 7-21). Bei einer Abfrage eines fertiggestellten Gerätes blinkt der Taster "Reparatur abgeschlossen", um den bestehenden Status hervorzuheben (Abb. 7-20 (1)). Die soeben beschriebenen Sicherheits-Mechanismen sollen die vollständige Eingabe aller Daten gewährleisten.

Reparaturdatenbank Version 1.3

Neue Eingabe **SPEICHERN** **Reset Eingabefenster** **Datensatz löschen** **Programm beenden**

Gerätenummer Neu

Suche Seriennummer: Suche Auftrag: GER: Datum Eingang:

Dieses Gerät war bereits mal zur Reparatur! Gerät Nr. von Es verbleiben noch Tage zur Fertigstellung!

Fehlerbeschreibung vom Kunden:
SFP Modul wird nicht erkannt! Link geht nicht up, kein Pegel am Ausgangspegel!

Tätigkeiten / Bemerkung:
FT, ET, SW- Update

Reparatur Zeit: h

Bearbeiter:

Repariert: ☒ Unreparabel: ☐ Ersatzgerät: ☐

Fehler Nr.	Fehlertext	Bauteilbenennung laut Bestückungsplan
352	Mech. defekter Bauteil (SMD-W)	L 11

Datum Ende:

voriger Datensatz **Reparatur abschließen** **Reparatur abgeschlossen** **nächster Datensatz** **NAT INST**

(1) (2)

Abb. 7-19: Abschluss einer Reparatur



Abb. 7-20: Reparatur abgeschlossen

Dieses Gerät war bereits mal zur Reparatur! Gerät Nr. von Es verbleiben noch Tage zur Fertigstellung!

Fehlerbeschreibung vom Kunden:
Alarm bei aufgesteckten (eingebauten) M...
ob auf MOD1 oder MOD2

Tätigkeiten / Bemerkung:

Reparatur Zeit: h

Bearbeiter:

Repariert: ☒ Unreparabel: ☐ Ersatzgerät: ☐

Fehler Nr.	Fehlertext	Bauteilbenennung laut Bestückungsplan
122	offene Lötst...	

Datum Ende:

voriger Datensatz **Reparatur abschließen** **Reparatur abgeschlossen** **nächster Datensatz** **NAT INST**

(1)

Abb. 7-21: Fehlermeldung

Alle Daten der ausgefüllten Felder werden in einem String zusammengefasst und gespeichert. Die bestehende Datei wird durch die neu generierte Datei ersetzt, welche den Status „Abgeschlossen“ beinhaltet und somit nicht mehr versehentlich verändert werden kann. Abbildung 7-25 zeigt den Dateiinhalt einer fertiggestellten Reparatur, welcher mittels Tabelle 7-3 tabellarisch erläutert wird.

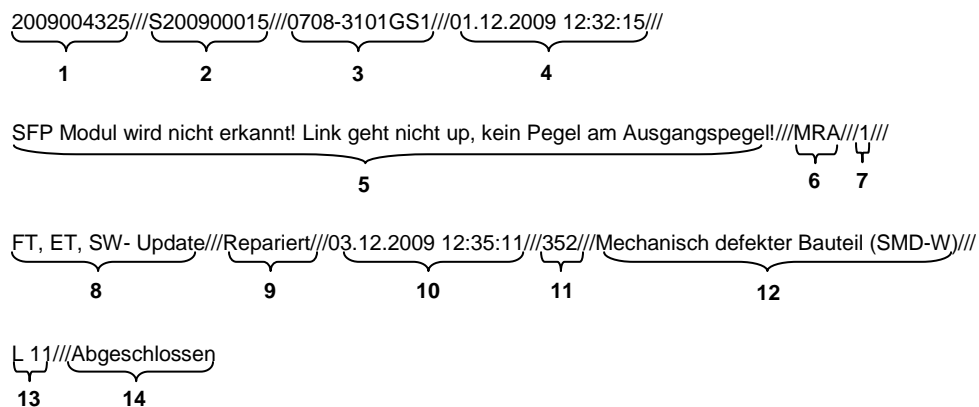


Abb. 7-25: Datensatz eines fertig reparierten Gerätes

1	Seriennummer des Gerätes
2	Auftragsnummer des Reparaturauftrages
3	Gerätename/Typ
4	Eingangs-/Erfassungsdatum der Reparatur
5	Fehlerbeschreibung des Kunden
6	Kurzzeichen Reparateur
7	Bearbeitungsdauer
8	Durchgeführte Arbeiten
9	Endstatus des Gerätes
10	Zeitpunkt der abschließenden Eingabe/Fertigstellung der Reparatur
11	Fehlercode
12	Fehlertext
13	Defekter Bauteil
14	Kennzeichnung für abgeschlossenen Datensatz

Tabelle 7-3 Inhalt einer Datei

7.4 Zusatzfunktionen

7.4.1 Neuanlage Gerätenummer und Gerätename

Über die Taste “Gerätenummer Neu” (Abb. 7-26 (1)) auf der Bedienermaske kann ein neuer Gerätetyp definiert werden. Durch Betätigen der Taste erscheinen zwei zusätzliche Eingabefelder (Abb. 7-26 (2)) und ein Schalter namens “Speichern” (Abb. 7-26 (3)).

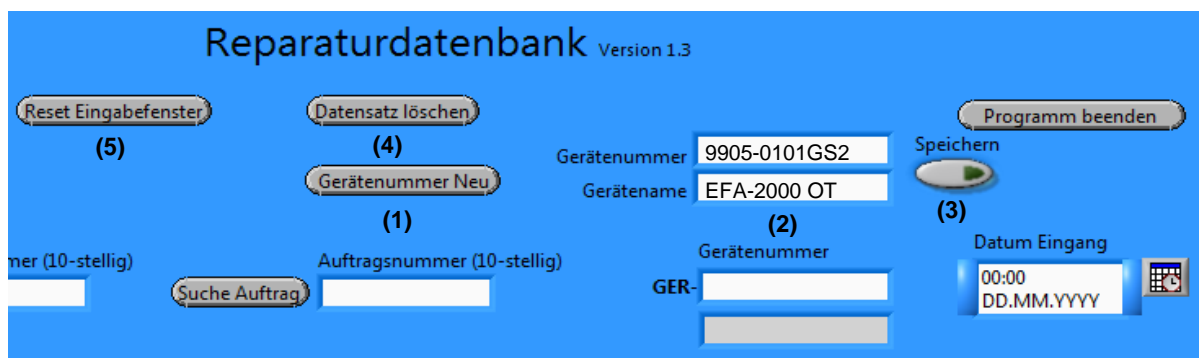


Abb. 7-26: Neuanlage Gerätetype

Hierbei kann die Gerätenummer und der dazugehörige Gerätename eingetragen und mit der Taste “Speichern” in die Textdatei “Gerätenummern” geschrieben werden. Durch erneutes Betätigen der Taste “Gerätenummer Neu” werden die Zusatzfelder sowie der Schalter wieder ausgeblendet.

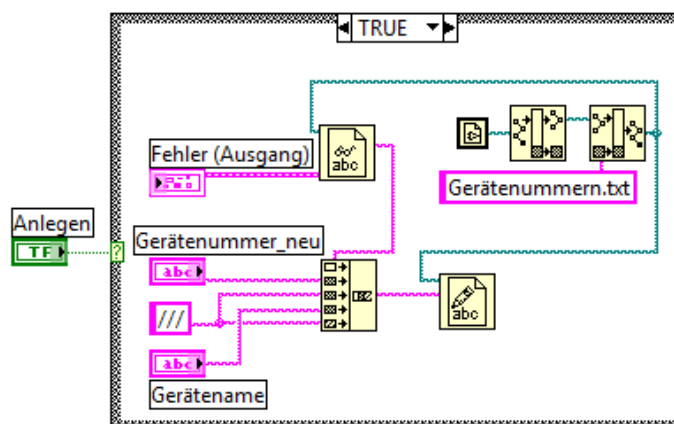


Abb. 7-27: Blockschaltbild Gerätenummernneuanlage

7.4.2 Löschen eines Datensatzes

Durch Betätigen des Tasters „Datensatz löschen“ (Abb. 7-26 (4)) ist es möglich, einen Datensatz unwiderruflich aus der Datenbank zu entfernen. Als Vorsichtsmaßnahme wird der Löschbefehl nicht sofort ausgeführt, sondern muss noch einmal bestätigt werden (Abb. 7-28).

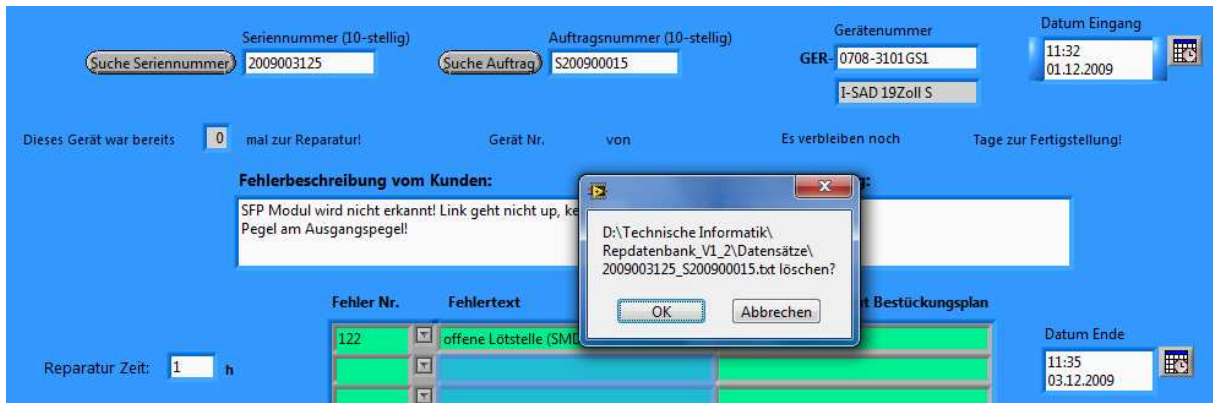


Abb. 7-28: Löschen eines Datensatzes

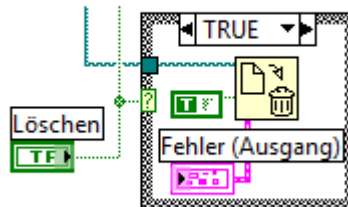


Abb. 7-29: Blockdiagramm (Ausschnitt) Löschen eines Datensatzes

7.4.3 Reset der Anzeige auf der Bedienoberfläche

Durch Drücken der Taste „Reset Eingabefenster“ (Abb. 7-26 (5)) werden alle Felder auf den Standardwert, also ein leeres Feld, zurückgesetzt. Es wird mit Hilfe von lokalen Variablen der jeweilige Ausgangswert in die Eingabefelder geschrieben. Boolesche Werte werden auf „False“ gesetzt, numerische auf „0“, Strings mit einem Leerzeichen beschrieben und Sichtbar-Eigenschaftsknoten auf unsichtbar gesetzt.

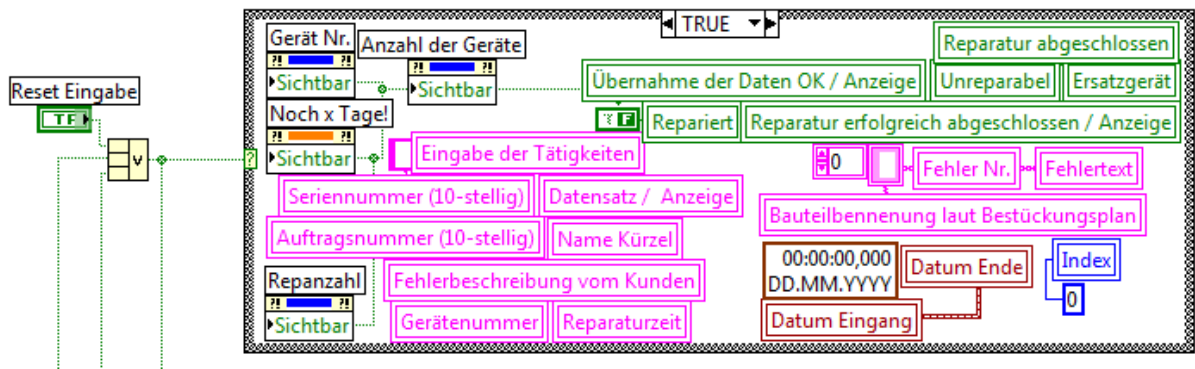


Abb. 7-30: Reset Eingabefenster

8 Zusammenfassung

8.1 *Diskussion der Ergebnisse*

Das Ziel der Diplomarbeit ist eine Reparaturdatenbank für eine elektronische Fertigung eines Mittelbetriebes zu erstellen.

Die theoretischen Grundlagen von Datenbanken und dem Programmiertool LabVIEW™ sowie der Dokumentation der erstellten Applikationen, bilden den schriftlichen Teil dieser Arbeit.

Als Ergebnis der praktischen Programmieraufgabe liegt eine mit LabVIEW™ realisierte Reparaturdatenbank vor, die sich derzeit in der Testphase befindet. Im Zuge dessen wurden bereits einige Verbesserungen und Zusatzapplikationen implementiert. Durch den Einsatz der Datenbank kann die Abarbeitung einzelner Reparaturaufträge effizienter durchgeführt werden. Zu jedem Gerät liegen die Datensätze, im Gegensatz zu handschriftlichen Aufzeichnungen oder geführten Excel-Tabellen, im selben Format mit für die Reparatur relevanten Daten vor.

8.2 *Zusammenfassung und Ausblick*

Die Datenbank kann mit einigen geringfügigen Anpassungen in Klein- und Mittelbetrieben, die sich mit der elektronischen Fertigung von Geräten befassen, eingesetzt werden. Hierzu kann eine sogenannte EXE-Datei erstellt werden, die mit der kostenlos von LabVIEW™ zur Verfügung gestellten "Run Time Engine" auf jedem Rechner, ohne den Erwerb einer Lizenz, ausgeführt werden kann.

Die erstellte Reparaturdatenbank ist seit Dezember 2009 in der Kundenreparaturabteilung in Siegendorf in Probebetrieb. Durch die Handhabung während der laufenden Testphase sind einige Erweiterungen der Datenbank vorgeschlagen worden.

Diese sind zum Beispiel:

- Drucken eines Lieferscheines über einen gesamten Auftrag
- Auswertung der Fehlerhäufigkeit eines bestimmten Bauteiles
- Auswertung der Reparaturanzahl in einem bestimmten Zeitraum
- Erweiterung der Datenbank für die Eingabe von Fertigungsreparaturen
- Generierung eines Fehlerreports über einen neu produzierten Auftrag

Literaturverzeichnis

2005 National Instruments Corporation: LabVIEW™- Grundlagen Version 8.0

2003-2006 National Instruments Corporation: Erste Schritte mit LabVIEW™ Version 8.20

Geisler, Frank: Datenbanken. Grundlagen und Design. 3., aktualisierte und erweiterte Auflage. Heidelberg, München, Landsberg, Frechen, Hamburg: mitp-Verlag, 2009

ISBN: 978-3-8266-5529-6

Georgi, Wolfgang & Metin, Ergun: Einführung in LabVIEW. 4., neu bearbeitete Auflage. München: Carl Hanser Verlag, 2009

ISBN: 978-3446-41560-7

Wikipedia 3.Jänner 2010

Quelle: <http://de.wikipedia.org/wiki/LabVIEW>

Abbildungsverzeichnis

Sofern nicht anders angegeben handelt es sich um eigene Abbildungen.

Abb. 3-1:	Das DBMS verwaltet den Zugriff auf die Datenbank (vgl. Geisler, 2009, S. 24)	9
Abb. 3-2:	Inhalt einer Datei (vgl. Geisler, 2009, S. 32)	9
Abb. 3-3:	Das Datenbanksystem (vgl. Geisler, 2009, S. 46)	10
Abb. 3-4:	Funktionen eines DBMS (vgl. Geisler, 2009, S. 51).....	13
Abb. 3-5:	Hierarchische-Datenbank	16
Abb. 3-6:	Netzwerk-Datenbank	17
Abb. 3-7:	Verbindung zwischen zwei Tabellen über Primär- /oder Fremdschlüssel (vgl. Geisler, 2009, S. 63).....	17
Abb. 3-8:	1:1 Beziehung zwischen Kunde und Adresse (vgl. Geisler, 2009, S. 69).....	18
Abb. 3-9:	1:N Beziehung zwischen Bearbeiter und mehreren Kunden (vgl. Geisler, 2009, S. 69)	18
Abb. 3-10:	M:N Beziehung zwischen Bearbeiter und Know-How (vgl. Geisler, 2009, S. 69)	18
Abb. 3-11:	Eine einfache Klassenhierarchie (vgl. Geisler, 2009, S. 74).....	19
Abb. 4-1:	Frontpanel.....	21
Abb. 4-2:	Blockdiagramm	21
Abb. 4-3:	Symbol- und Anschlussfeld.....	22
Abb. 4-4:	Beispiel VI mit Frontpanel und dazugehörigem Blockdiagramm	23
Abb. 4-5:	Farbcodierung verschiedener Datentypen	24
Abb. 4-6:	Bedien- und Anzeigeelemente-Palette.....	25
Abb. 4-7:	Beispiele von möglichen Anzeige- und Bedienelemente	25
Abb. 4-8:	Anzeige- und Bedienelemente (zu Abb. 4-7) im Blockdiagramm.	26
Abb. 4-9:	Funktionenpalette	26
Abb. 4-10:	Ein kleiner Teil der Strukturelemente und Funktionen.....	27
Abb. 4-11:	Werkzeugpalette mit Beispielen	27
Abb. 5-1:	Typenschild mit Barcode.....	28

Abb. 6-1:	Datenbankstruktur.....	29
Abb. 6-2:	VI-Hierarchie.....	30
Abb. 7-1:	Desktop Icon.....	32
Abb. 7-2:	Eingabemaske.....	32
Abb. 7-3:	Felder für Neuanlage ausgefüllt.....	33
Abb. 7-4:	Fehlermeldung.....	34
Abb. 7-5:	Dateiname und Dateiinhalt einer Neueingabe.....	34
Abb. 7-6:	Funktion leerer String, Funktion String-Länge.....	35
Abb. 7-7:	Blockdiagramm Neuanlage (Ausschnitt).....	35
Abb. 7-8:	Blockdiagramm Gerätenummernanzeige.....	36
Abb. 7-9:	Blockdiagramm Fehlerausgabe.....	36
Abb. 7-10:	Seriennummernsuche.....	37
Abb. 7-11:	Blockdiagramm Seriennummernsuche.....	38
Abb. 7-12:	Blockdiagramm Suchfunktion.....	38
Abb. 7-13:	Blockdiagramm String in Datumsanzeige.....	39
Abb. 7-14:	Blockdiagramm Restzeitberechnung.....	39
Abb. 7-15:	Suche nach Auftrag.....	40
Abb. 7-16:	Blockdiagramm Auftragsnummernsuche.....	40
Abb. 7-17:	Vollständig ausgefüllte Eingabemaske.....	41
Abb. 7-18:	Bestückungsplan eines Gerätes.....	42
Abb. 7-19:	Abschluss einer Reparatur.....	45
Abb. 7-20:	Reparatur abgeschlossen.....	45
Abb. 7-21:	Fehlermeldung.....	45
Abb. 7-22:	Blockdiagramm Fehlereingabe.....	46
Abb. 7-23:	Blockdiagramm Fehlertextauslesen.....	47
Abb. 7-24:	Blockdiagramm Fehlerausgabe Abschluss.....	47
Abb. 7-25:	Datensatz eines fertig reparierten Gerätes.....	48
Abb. 7-26:	Neuanlage Gerätetype.....	49
Abb. 7-27:	Blockschaltbild Gerätenummernneuanlage.....	49
Abb. 7-28:	Löschen eines Datensatzes.....	50
Abb. 7-29:	Blockdiagramm (Ausschnitt) Löschen eines Datensatzes.....	50
Abb. 7-30:	Reset Eingabefenster.....	51

Tabellenverzeichnis

Sofern nicht anders angegeben handelt es sich um selbst erstellte Tabellen.

Tabelle 6-1:	Kurzbeschreibung VIs und Sub-VIs	31
Tabelle 7-1:	Dateiinhalte einer Neuanlage	35
Tabelle 7-2:	Fehlernummern mit Klartext	43
Tabelle 7-3:	Inhalte einer Datei	48

Anhang

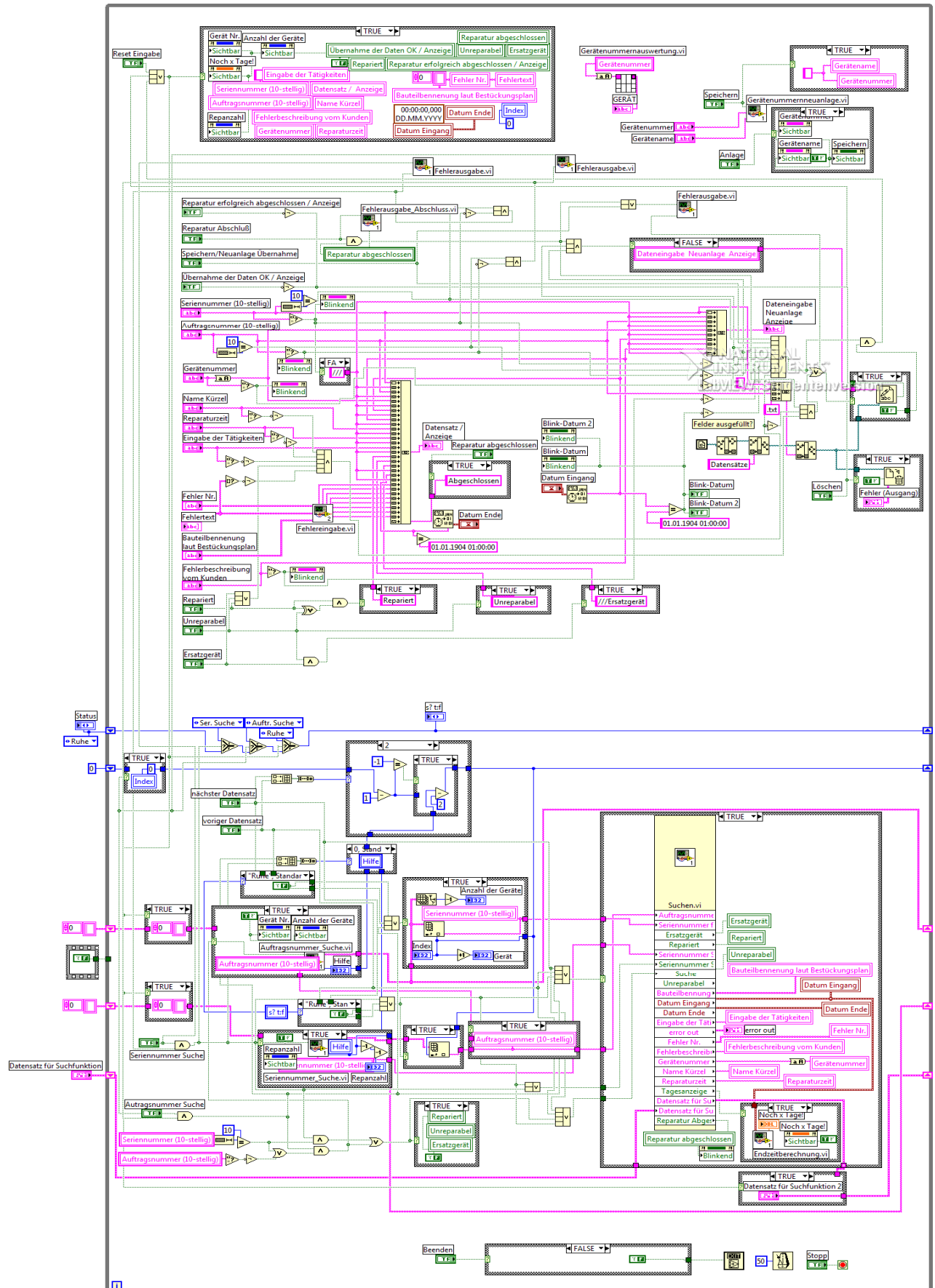
Anmerkung:

Einige abgebildete Blockdiagramme zeigen jeweils nur einen Programmausschnitt, da die einzelnen Strukturen wie z. B. Case-Strukturen oder Sequenzen zwei bzw. mehrere Fenster, die nicht gleichzeitig anzeigbar sind, enthalten.

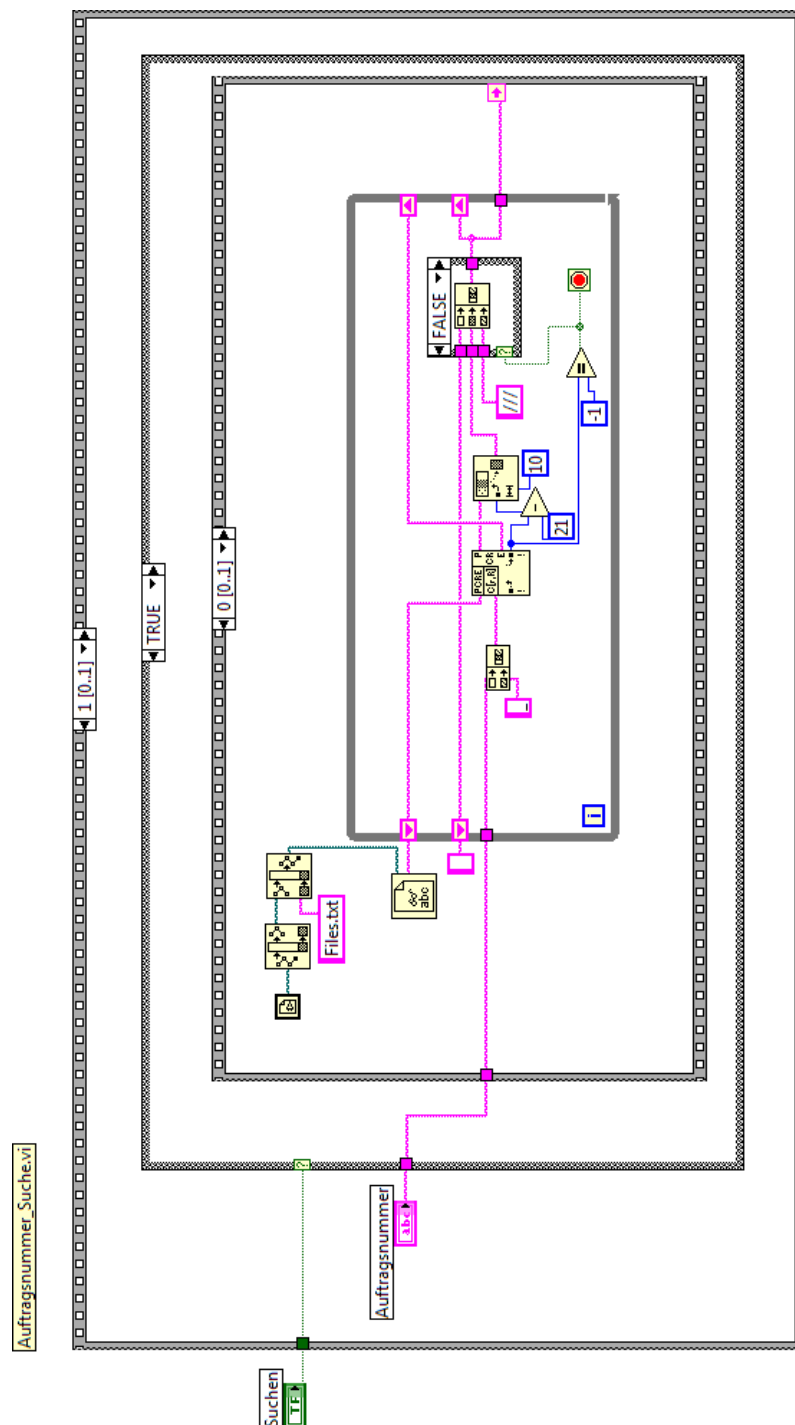
Anhang A Benutzeroberfläche

[illegible]

Anhang B Blockdiagramm Benutzeroberfläche

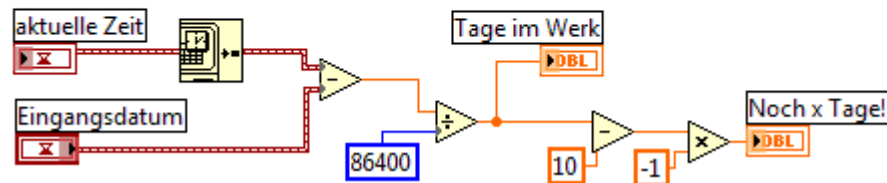


Anhang C Blockdiagramm Auftragsnummer Suche



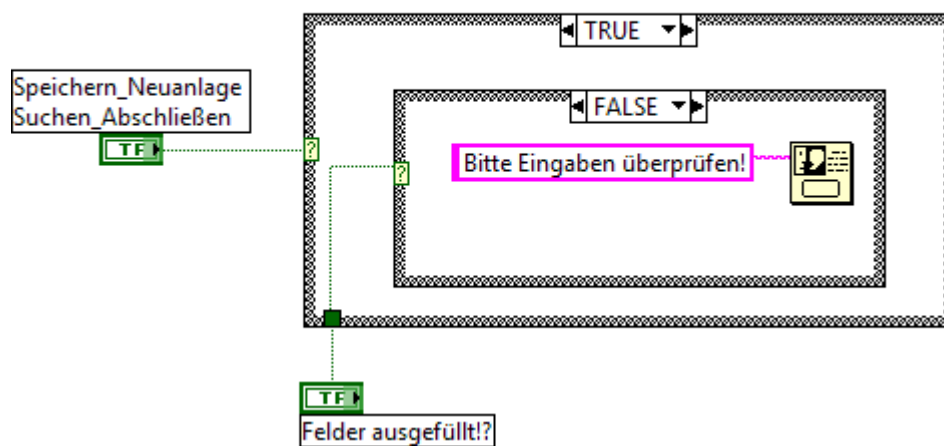
Anhang D Blockdiagramm Endzeitberechnung

Endzeitberechnung.vi

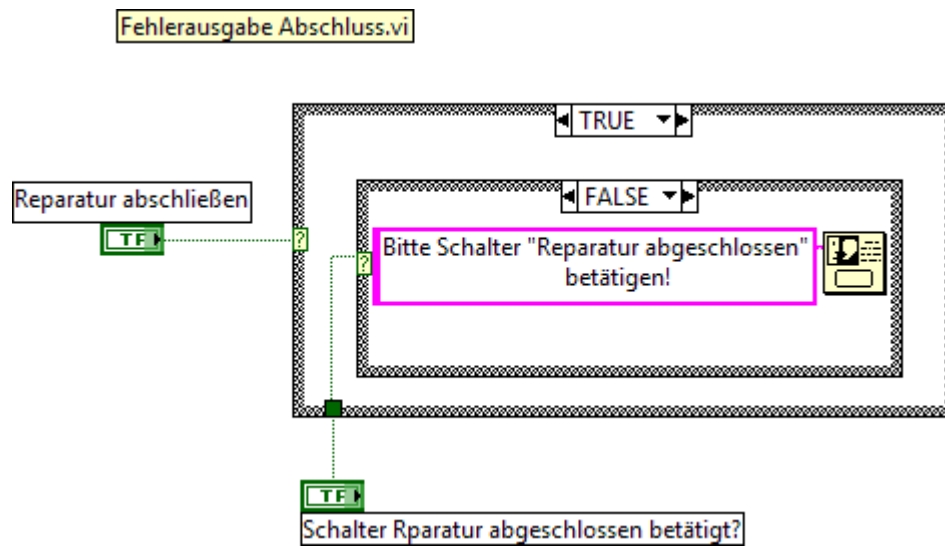


Anhang E Blockdiagramm Fehlerausgabe

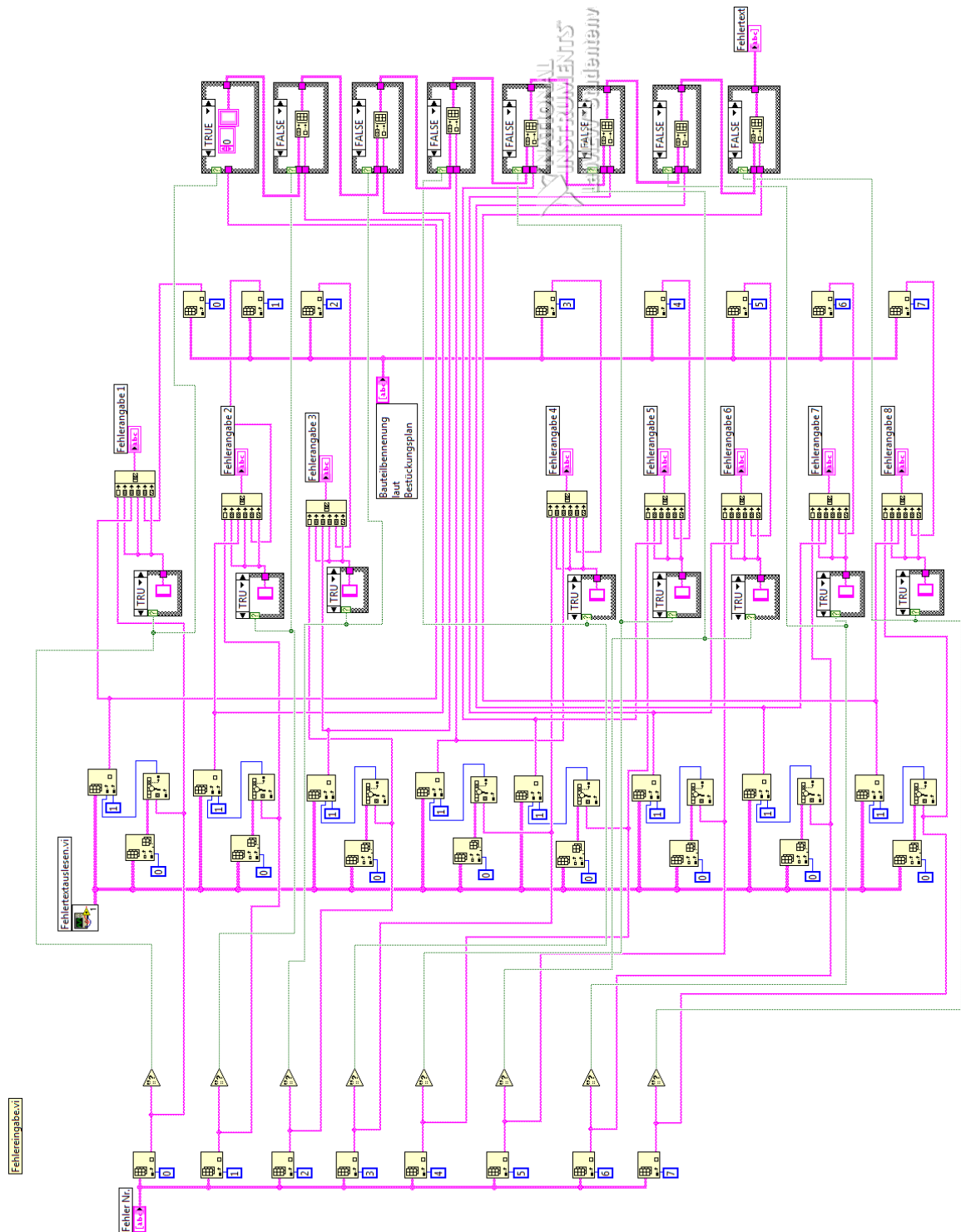
Fehlerausgabe.vi



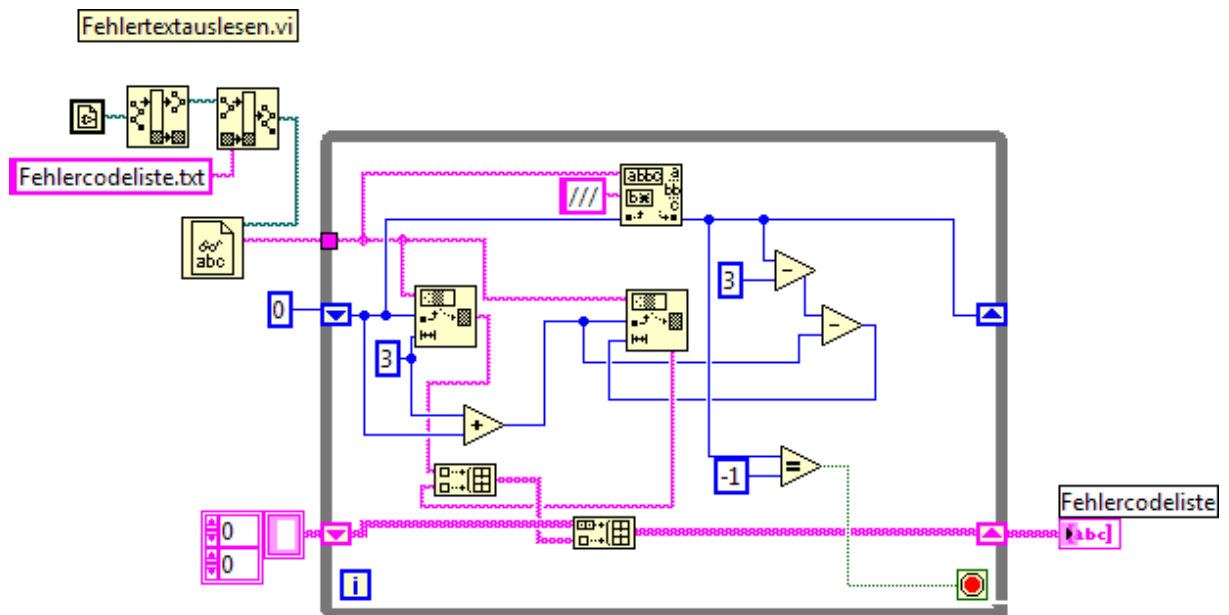
Anhang F Blockdiagramm Fehlerausgabe Abschluss



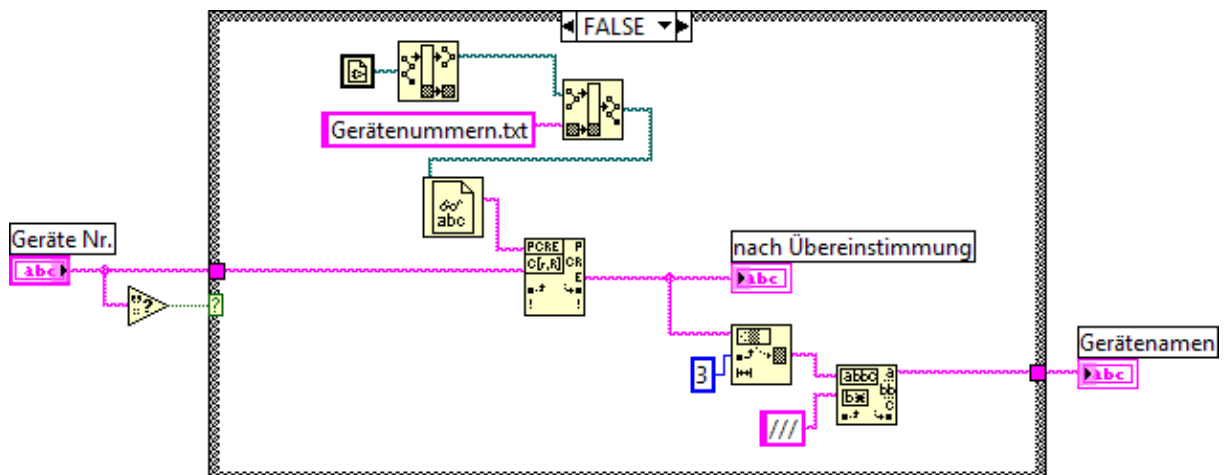
Anhang G Blockdiagramm Fehlereingabe



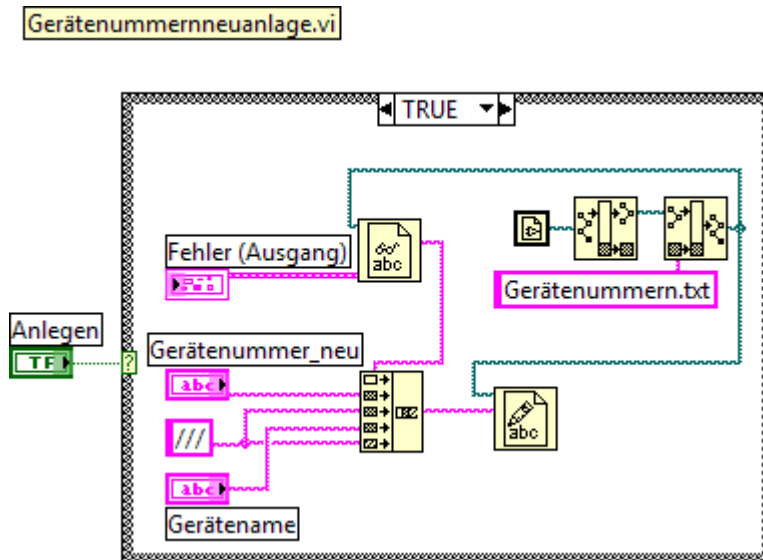
Anhang H Blockdiagramm Fehlertext auslesen



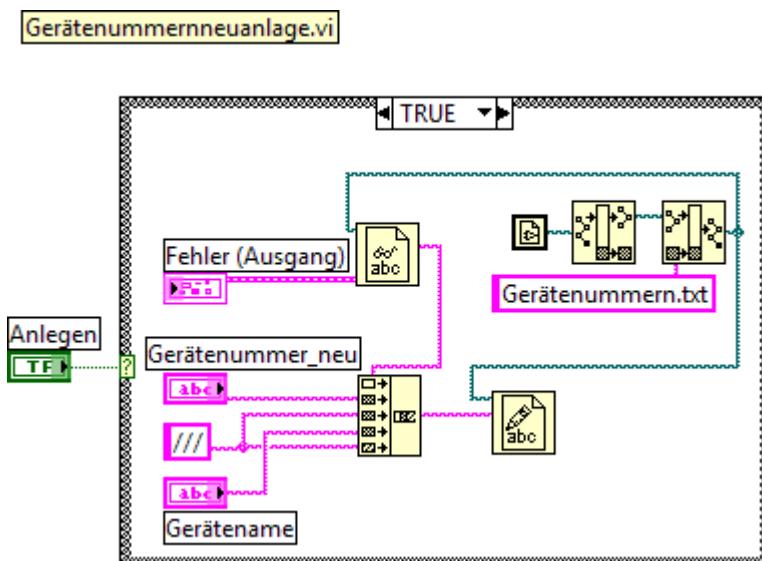
Anhang I Blockdiagramm Gerätemummernauswertung



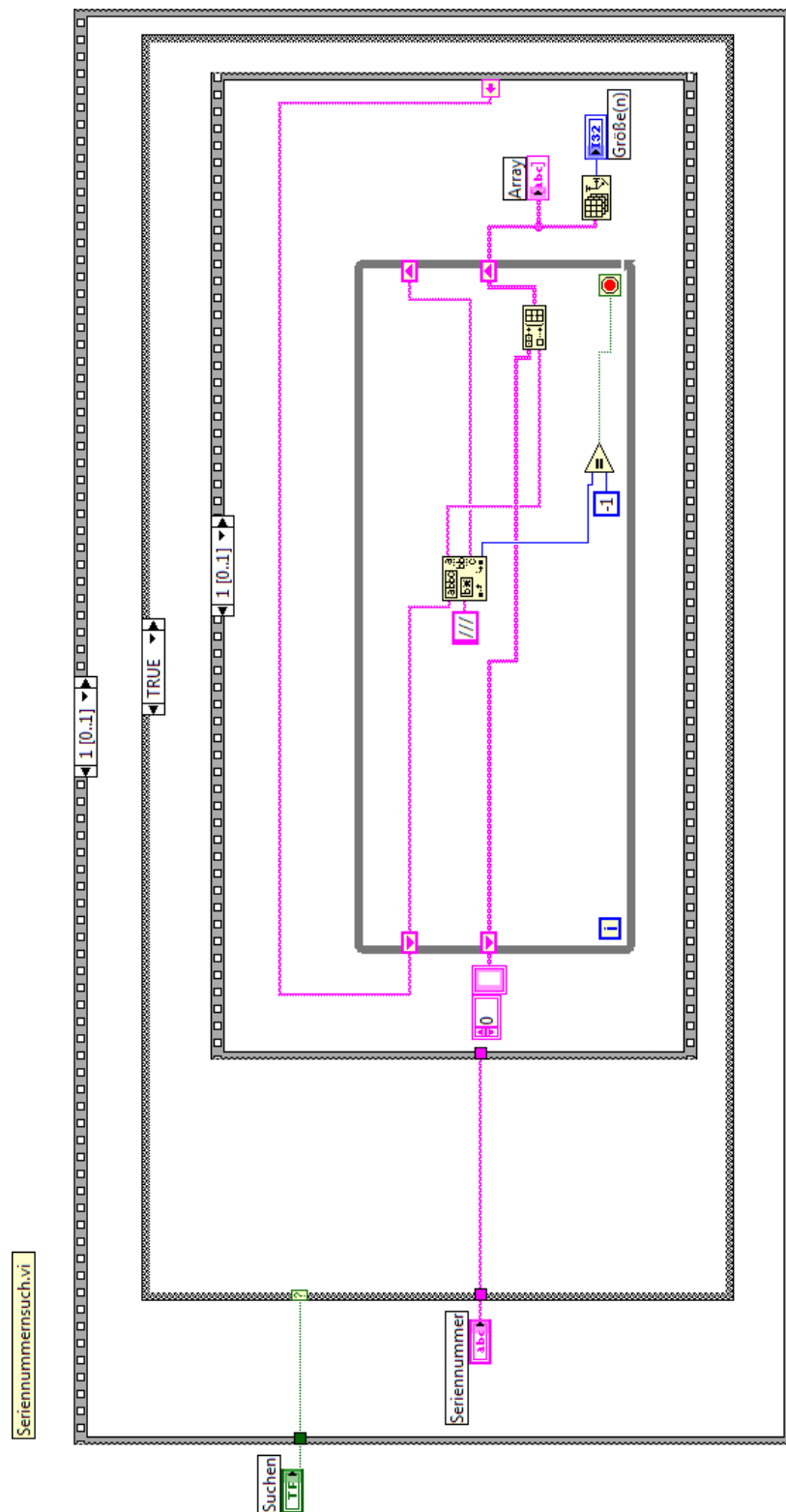
Anhang J Blockdiagramm Gerätenummernauswertung



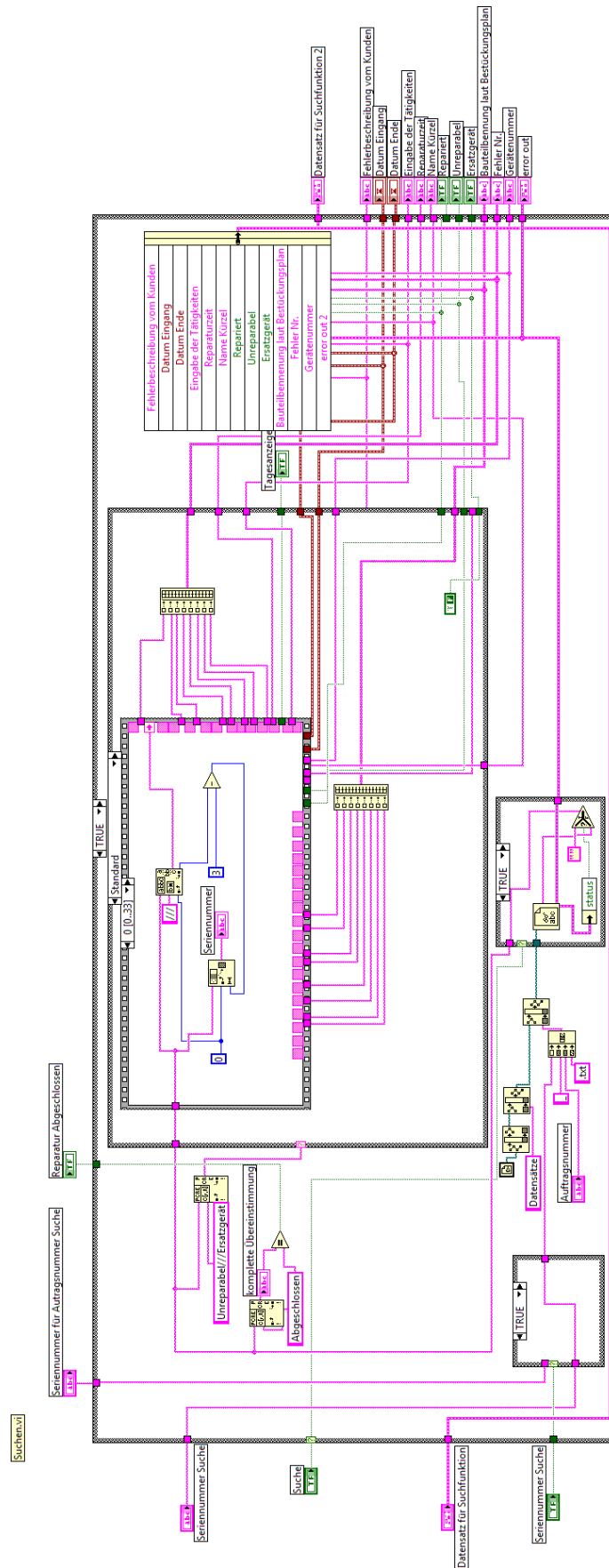
Anhang K Blockdiagramm Gerätenummernneuanlage



Anhang L Blockdiagramm Seriennummernsuche



Anhang M Blockdiagramm Suchen



Anhang N Blockdiagramm Zeitberechnung

